महाराष्ट्र राज्य तंत्र शिक्षण मंडळ, मुंबई

(स्वायत्त) (ISO 9001:2015) (ISO/IEC 27001:2013)

अभियांत्रिकी आणि तंत्रज्ञान पदविका

शिक्षण पुस्तिका

(Learning Material)

EMBEDDED SYSTEM USING ‘C’

(314338)

यंत्र अभियांत्रिकी गट

मराठी – इंग्रजी (द्विभाषिक) माध्यम

(अभियांत्रिकी व तंत्रज्ञान चतुर्थ सत्र पदविका)

MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION MUMBAI

शिक्षण पुस्तिका

(Learning Material)

एम्बेडेड सिस्टिम युजिंग ‘C’

EMBEDDED SYSTEM USING ‘C’

(314338)

मराठी-इंग्रजी द्विभाषिक माध्यम

(अभियांत्रिकी व तंत्रज्ञानातील चतुर्थ सत्र पदविका)

महाराष्ट्र राज्य तंत्र शिक्षण मंडळ, मुंबई

(स्वायत्त) (ISO 9001:2015) (ISO/IEC 27001:2013)

एम.एस. बी. टी. ई.
मार्गदर्शक

श्री. एस. एस. हरीप

निवड श्रेणी अधिव्याख्याता, यांत्रिकी अभियांत्रिकी

संकलक

कु. एम. जी. सौन्दत्ते

अधिव्याख्याता, अणुविद्‌युत आणि संगणक अभियांत्रिकी

कु. के. डी. जोंग

अधिव्याख्याता, अणुविद्‌युत आणि संगणक अभियांत्रिकी

कु. एस. एस. नांदणे

अधिव्याख्याता, अणुविद्‌युत आणि संगणक अभियांत्रिकी

सौ. ए. एस. धड्डे

अधिव्याख्याता, अणुविद्‌युत आणि संगणक अभियांत्रिकी

कु. एस. आर. खिलारे

अधिव्याख्याता, अणुविद्‌युत आणि संगणक अभियांत्रिकी

समीक्षक

सौ. ए. एन. नाईक

अधिव्याख्याता, मेकॅट्रॉनिक्स

मुख्य समन्वयक

श्री. बी. एस. ताशिलदार

प्राचार्य

संस्था समन्वयक

श्री.एस. एस. कोथळे

विभागप्रमुख, मेकॅट्रॉनिक्स

MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION * MUMBAI

महाराष्ट्र राज्य तंत्र शिक्षण मंडळ

(स्वायत्त) (ISO: ९००१:२०१५) (ISO/IES: २७००१-२०१३)

शासकीय तंत्रनिकेतन इमारत, चौथा मजला, ४९, खेरवाडी, बांद्रा (पूर्व), मुंबई - ४०० ०५१.

दूरध्वनी क्र.: ०२२-६२५४२१७० / १६१

Email : director@msbte.com Web : www.msbte.org.in

प्रास्ताविक

महाराष्ट्र राज्यातील पदविका स्तरावरील तंत्रशिक्षणामध्ये विद्यार्थ्यांचे रोजगार कौशल्य विकसित करून विद्यार्थ्यांचा सर्वांगीण विकास घडवून आणण्याकरिता महाराष्ट्र राज्य तंत्रशिक्षण मंडळ कटिबद्ध आहे. उद्योगधंद्यातील बदलत्या तंत्रज्ञानाशी संबंधित गरजा लक्षात घेऊन महाराष्ट्र राज्य तंत्र शिक्षण मंडळाकडून पदविका अभ्यासक्रम वेळोवेळी अद्यावत करण्यात येतो . अभियांत्रिकी पदविका अभ्यासक्रम शिकत असतांना संकल्पनात्मक ज्ञान, सुसंगत संदर्भ, प्रश्न विचारणे, विश्वसनिय पुरावे, कारणमीमांसा आणि सुस्पष्ट निकष यांचा वापर करून अर्थाची उकल करण्याची, विश्लेषण व मूल्यमापन करण्याची तसेच तर्काने अनुमान काढण्याची क्षमता म्हणजेच चिकित्सक विचार विद्यार्थ्यांमध्ये अधिक दृढ होतील असा मला विश्वास आहे. जेव्हा विद्यार्थी ज्ञान मिळवण्याच्या माध्यमाशी पूर्णपणे परिचित आणि सोयीस्कर असतात, तेव्हा त्यांच्यासाठी वर्गातील चर्चेत भाग घेणे सोपे होते, संकल्पनात्मक व सैद्धांतिक बाबींचे आकलन परिपूर्ण होते, संज्ञानात्मक क्षमता सुधारते आणि त्यांचा आत्मविश्वास देखील वाढतो या सर्व गोष्टींचा विचार करून मंडळाकडून शैक्षणिक सामुग्रीची निर्मिती करण्यात आलेली आहे . भारत देश हा खेड्यापाड्यातून विकसित झालेला देश असून ग्रामीण भागातील विद्यार्थ्यांना तांत्रिक शिक्षण घेतांना भाषेचा अडसर न येता तांत्रिक बाबींचा आशय समजून घेणे शक्य होईल या दृष्टिकोनातून महाराष्ट्र राज्य तंत्र शिक्षण मंडळाने पदविका स्तरावरील तांत्रिक शिक्षणाकरीता विद्यार्थ्यांना मराठी-इंग्रजी द्विभाषिक माध्यमाचा पर्याय शैक्षणिक वर्ष २०२१-२२ पासून उपलब्ध करून दिलेला आहे .

राष्ट्रीय शैक्षणिक धोरण-२०२० प्रादेशिक भाषेतील शिक्षणास प्रोत्साहन देते, ज्यामुळे विद्यार्थ्यांना तांत्रिक अभ्यासक्रमांसाठी प्रादेशिक भाषांतुन शिक्षणाचे माध्यम निवडता येते. सदर धोरणामुळे प्रादेशिक भाषांमध्ये तांत्रिक सामग्री आणि अभ्यास सामग्रीचा विकास आणि भाषांतर निर्माण करण्याची आवश्यकता आहे . त्यास अनुसरून मंडळाने मराठी-इंग्रजी द्विभाषिक माध्यमाचा पर्याय द्वितीय व तृतीय वर्षाकरिताही उपलब्ध करून देण्यात आला आहे. तसेच त्याकरिताची शैक्षणिक सामग्रीही संबंधीत भागधारकरांना उपलब्ध करून देण्यात येत आहे.

पदविका स्तरावरील तंत्रशिक्षण अधिक दर्जेदार करण्यासाठी महाराष्ट्रातील अनुभवी व तज्ञ अध्यापकांनी व्यवहारिक मराठी भाषा व इंग्रजी भाषेतील तांत्रिक शब्दावली यांचा वापर करून मराठी - इंग्रजी भाषेचा सुवर्णमध्य साधण्याचा प्रयत्न केलेला आहे. मंडळाच्या स्तरावर गठीत सुकाणू समितीमार्फत सदर शैक्षणिक सामुग्रीचा दर्जा, तसेच इतर बाबींची तपासणी करण्यात आलेली आहे. त्यामुळे सदर शैक्षणिक सामुग्री अधिक सम्पन्न झालेली असून विद्यार्थी त्यांच्या व्यक्तिमत्त्वाचा सुसंवादी आणि सर्वांगीण विकास साधतील. परिणामतः विश्वस्तरीय मनुष्यबळाच्या गरजा पूर्ण करण्यात महाराष्ट्र राज्य अग्रेसर राहील व पर्यायाने राष्ट्रनिर्मिती करीता निश्चितच हातभार लागेल , असा मला विश्वास आहे.

अभियांत्रिकी पदविका अभ्यासक्रमातील प्रमुख विषयांची मराठी-इंग्रजी द्विभाषिक शैक्षणिक सामुग्री बनविण्यासाठी अध्यापक व सुकाणू समितीचे सदस्य यांनी दर्शविलेले समर्पण व वचनबद्धता कौतुकास पात्र आहे , या सर्वांचे मी मनः पूवक अभिनंदन करतो !

(प्रमोद नाईक)
संचालक
म. रा. तंत्र शिक्षण मंडळ, मुंबई.

अनुक्रमणिका

अ. क्र	युनिटचे नाव	पान क्र.
1	एम्बेडेड सिस्टमचा आढावा (Overview of Embedded System)	1
2	8051 प्रोग्रॅमिंग युजिंग एम्बेडेड 'C' (8051 Programming using Embedded ‘C’)	11
3	कम्युनिकेशन स्टँडर्ड आणि प्रोटोकॉल्स (Communication Standards and Protocols)	32
4	8051 मायक्रोकंट्रोलरसह इनपुट आणिआउटपुट डिव्हाइसेसचे इंटरफेसिंग (Interfacing of Input and Output Devices with 8051)	50
5	8051 मायक्रोकंट्रोलरचे एप्लीकेशन (Applications of 8051 Microcontroller)	68

MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION * MUMBAI

युनिट 1. एम्बेडेड सिस्टमचा आढावा
(Overview of Embedded System)

Course Outcome: - Classify the different types of embedded systems.

सिद्धांत शिक्षण निष्पत्ती (Theory Learning Outcomes)

TLO 1.1 एम्बेडेड प्रणालीच्या वैशिष्ट्यांसह तिच्या ब्लॉक आकृतीचे स्पष्टीकरण द्या.
TLO 1.2 एम्बेडेड प्रणालींचे वर्गीकरण करा.
TLO 1.3 वॉन-न्यूमन आणि हार्वर्ड आर्किटेक्चर यांच्यातील तुलना व भेद स्पष्ट करा.
TLO 1.4 8051 मायक्रोकंट्रोलरची वैशिष्ट्ये व पिन संरचना स्पष्ट करा.
TLO 1.5 पॉवर सेविंग ऑपशन्स (Power-saving options) स्पष्ट करा.

1.1 एम्बेडेड सिस्टम:

एम्बेडेड सिस्टम ही एक मायक्रोप्रोसेसर किंवा मायक्रोकंट्रोलर आधारित प्रणाली आहे जी एक करण्यासाठी डिझाइन केलेली आहे एम्बेडेड सिस्टम ही एक किंवा दोन विशिष्ट कार्यांसाठी डिझाइन केलेली समर्पित संगणक प्रणाली आहे. ही प्रणाली संपूर्ण उपकरण प्रणालीचा एक भाग म्हणून एम्बेड केलेली आहे ज्यात हार्डवेअर समाविष्ट आहे, जसे की इलेक्ट्रिकल आणि यांत्रिक घटक.

एम्बेडेड सिस्टीम हे हार्डवेअर आणि सॉफ्टवेअरचे संयोजन आहे जेथे सॉफ्टवेअर हे सहसा फर्मवेअर म्हणून ओळखले जाते जे हार्डवेअरमध्ये एम्बेड केलेले असते.

एम्बेडेड सिस्टमची उदाहरणे

एम्बेडेड सिस्टीमची काही उदाहरणे म्हणजे MP3 प्लेयर, मोबाईल फोन, व्हिडिओ गेम कन्सोल, डिजिटल कॅमेरे, डीव्हीडी प्लेयर आणि जीपीएस. घरगुती उपकरणे, जसे की मायक्रोवेव्ह ओव्हन, वॉशिंग मशीन आणि डिशवॉशर, फ्लेक्झिबिलिटी आणि एफिशियन्सी प्रोव्हाइड करण्यासाठीएम्बेडेड सिस्टम समाविष्ट करतात.

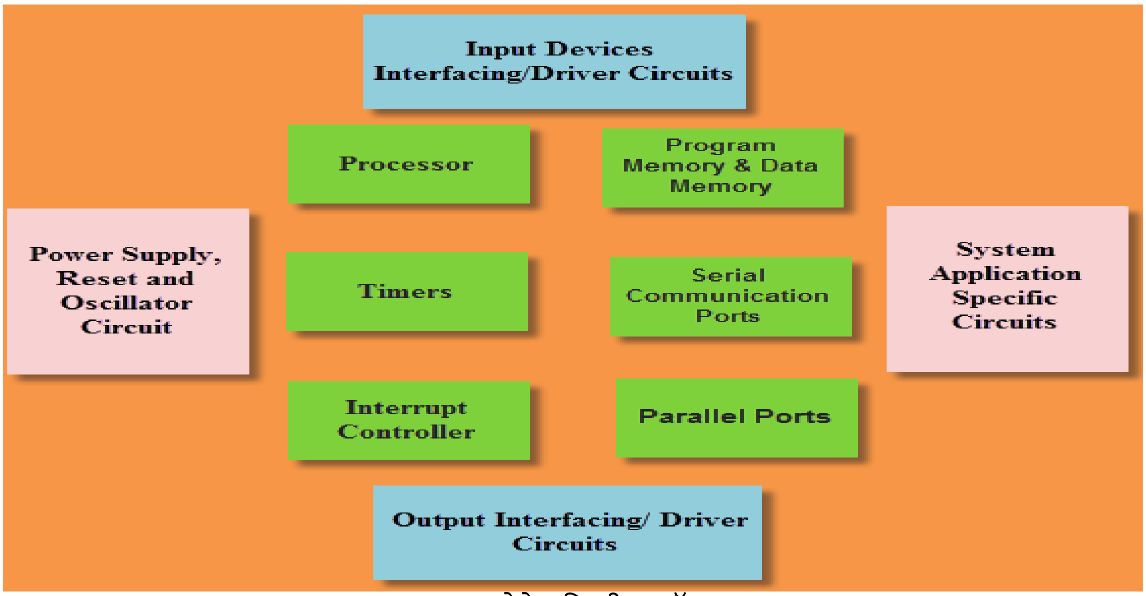


Figure1.1 एम्बेडेड सिस्टीम ब्लॉक डायग्राम

एम्बेडेड सिस्टीममध्ये खाली दिलेल्या आणि Figure1.1 मध्ये दाखवल्याप्रमाणे वेगवेगळे घटक एम्बेड केलेले असतात.

हार्डवेअर घटक:

- प्रोसेसर
- मेमरी

- पॉवर सप्लाय
- टाइमर/इंटरप्ट्स
- इनपुट/आउटपुट सर्किट्स
- सिरियल आणि पॅरलल कम्युनिकेशन पोर्ट
- सिस्टम ॲप्लिकेशन स्पेसिफिक सर्किट्स(SASC)

एम्बेडेड प्रोसेसर: हेएम्बेडेड सिस्टमचेहार्ट आहे. त्यात दोन आवश्यक युनिट्स आहेत: कंट्रोल युनिट आणि एक्झिक्युशन युनिट. कंट्रोल युनिट मेमरीमधून इन्स्ट्रक्शन मिळवते आणि एक्झिक्युशन युनिटमध्ये प्रोग्राम कंट्रोल टास्कसाठी सूचनांची अंमलबजावणी करण्यासाठी ॲरेथमिटिक आणि लॉजिक युनिट आणि सर्किट्स समाविष्ट असतात.

पॉवर सप्लाय, रीसेट आणि ऑसिलेटर सर्किट:

बहुतेक सिस्टम्सचा स्वतःचा पॉवर सप्लाय असतो. काही एम्बेडेड सिस्टम्सचा स्वतःचा पॉवर सप्लाय नसतो. या एम्बेडेड सिस्टम्स बाह्य पॉवर सप्लायवापरतात. उदा. USB आधारित एम्बेडेड सिस्टम, नेटवर्क इंटरफेस कार्ड, ग्राफिक्स एक्सीलरेटर इत्यादी पीसी मधूनपॉवर सप्लाय घेतात.

रिसेट म्हणजे प्रोसेसर पॉवर अपवर प्रोग्राम काउंटरमध्ये डीफॉल्टनुसार सेट केलेल्या सुरुवातीच्या ॲड्रेस पासून इन्स्ट्रक्शन प्रोसेस करण्यास सुरुवात करतो. क्लॉक सर्किट इन्स्ट्रक्शनच्या अंमलबजावणीचा वेळ, CPU मशीन सायकल नियंत्रित करते.

टाइमर:

टाइमर सर्किट सिस्टम क्लॉक किंवा RTC (रिअल टाइम क्लॉक) म्हणून योग्यरित्या कॉन्फिगर केलेले आहे. विविध कार्ये शेड्यूल करण्यासाठी आणि रिअल टाइम प्रोग्रामिंगसाठी RTC (रिअल टाइम क्लॉक) किंवा सिस्टम क्लॉक आवश्यक आहे.

प्रोग्राम आणि डेटा मेमरी:

बहुतेक एम्बेडेड प्रोसेसरमध्ये प्रोग्राम आणि डेटा साठवण्यासाठी ROM, RAM, फ्लॅश/EEPROM, EPROM/PROM सारखी अंतर्गत मेमरी असते.

इंटरप्ट कंट्रोलर: ही एक इंटरप्ट हँडलिंग मेकॅनिझम आहे जी विविध प्रक्रियांमधील इंटरप्ट हाताळण्यासाठी आणि सर्विससाठी एकाच वेळी पेंडिंगअसलेल्या अनेक इंटरप्ट हाताळण्यासाठी एम्बेडेड सिस्टममध्ये अस्तित्वात असणे आवश्यक आहे.

इनपुट/ आउटपुट (I/O) पोर्ट:

I/O पोर्ट सेन्सर, की बटणे, ट्रान्सड्यूसर, LEDs, LCD ॲक्ट्युएटर, अलार्म, मोटर्स, व्हॅल्यूज, प्रिंटर इत्यादी बाह्य उपकरणांना इंटरफेस करण्यासाठी वापरले जातात. पोर्टचे दोन प्रकार आहेत, पार्लरलिझम आणि सिरीयल पोर्ट. पॅरलल पोर्ट कमी अंतराच्या कम्युनिकेशनमध्ये वापरले जातात तर सिरीयल पोर्ट लांब अंतराच्या कम्युनिकेशनमध्ये वापरले जातात.

इनपुट आणि आउटपुट डिव्हाइस इंटरफेसिंग/ड्रायव्हर सर्किट:

मोटर्स, अक्चुएटर, व्हॉल्व्ह, सेन्सर सारखी काही I/O उपकरणे प्रोसेसरशी सुसंगत नाहीत. म्हणूनच, आय/ओ इंटरफेस सर्किट्स एम्बेडेड प्रोसेसरशी जोडलेल्या इनपुट आणि आउटपुट डिव्हाइसेसना चालविण्यासाठी डिझाइन केलेले आहेत.

सिस्टम ॲप्लिकेशन स्पेसिफिकसर्किट्स:

हे असे सर्किट्स आहेत जे विशिष्ट टारगेट सर्किट्स नियंत्रित करू शकतात. त्यामध्ये एडीसी(ADC), डीएसी(DAC), रिले, सेन्सर्स इत्यादी असतात.

1.2 एम्बेडेड सिस्टीमची वैशिष्ट्ये:

1) प्रोसेसर पॉवर: प्रोसेसरची निवड आवश्यक असलेल्या रजिस्टर साईजकाम करण्यासाठी किती प्रोसेसिंग पॉवरची आवश्यकता आहे यावर आधारित असते.8 बिट, 16 बिट, 32बिट आणि 64 बिट मायक्रोकंट्रोलर दिलेले आहेत. वेगवेगळ्या मायक्रोकंट्रोलरसाठी प्रोसेसिंग पॉवर वेगळी असते. हायक्लॉक, स्पीड आणि अड्रेससिंग देण्यास सक्षम मायक्रोकंट्रोलर उपलब्ध आहेत. ऑडिओआणि व्हिडिओ सिग्नलच्या रिअलटाइम अनॅलिसिससाठी खूप पॉवरफुल डीएसपी उपलब्ध आहेत.

2) मेमरी: एम्बेडेड सिस्टमसाठी विकसित केलेले प्रोग्राम फर्मवेअर म्हणून मानले जातात आणि रॉम (ROM)किंवा फ्लॅश मेमरीमध्ये स्टोअरकेले जातात. एम्बेडेड सिस्टम मर्यादित रिसोर्सेसवरचालत असल्याने, कमी किमतीत कमी मेमरी पुरेशी असते.

3)ऑपरेटिंग सिस्टम: एम्बेडेड ओएस रिलायबल आणि मेमरी, साईज टाईमआणि प्रोसेसिंग पॉवरवर टाईटमर्यादांसह लण्यास सक्षम असणे आवश्यक आहे. बहुतेक एम्बेडेडऑपरेटिंगसिस्टिमयाआरटीओएस (Real Time Operating System) आहेत.

4)रिलायबिलिटी: एम्बेडेड सिस्टीम बहुतेकदाअशामशीनमध्येअसते ज्या वर्षानुवर्षे त्रुटींशिवाय सतत चालतील अशी अपेक्षा असते आणि काही सिस्टिममध्ये त्रुटीआढळल्यास ती स्वतःहून रिपेअर होते. म्हणून, सॉफ्टवेअर सामान्यतः वैयक्तिक संगणकांपेक्षाअधिक काळजीपूर्वक विकसित आणि चाचणी केली जाते आणि डिस्कड्राइव्ह, स्विचेस किंवा बटणे यांसारखे मूविंग पार्ट किंवा मेकॅनिकल पार्ट टाळले जातात.

5) युनिट कॉस्ट: एनआरई (NRE) खर्च वगळता, सिस्टमच्या प्रत्येक प्रतीच्या उत्पादनाचा आर्थिक खर्चअसतो.

6) एनआरई (नॉनरिकरींग इंजिनिअरिंग कॉस्ट): सिस्टम डिझाइन करण्याचा एक वेळचा खर्च ज्यामध्येसंशोधन आणि विकास आणि प्रथम कार्यात्मक आणि चाचणी केलेल्या विकासात गुंतवलेले पैसे समाविष्ट आहेत. एकदा सिस्टम डिझाइन झाल्यानंतर, कोणताही अतिरिक्त डिझाइन खर्च न घेता कितीही युनिट्स तयार करता येतात (म्हणूनच "नॉन-रिकरींग" हा शब्दवापरलाआहे).

7) साईज: एम्बेडेड सिस्टम आकारात शक्य तितकी लहान असावी जी सिस्टमला आवश्यक असलेल्या भौतिक जागेपेक्षा वेगळी नसावी, बहुतेकदा सॉफ्टवेअरसाठी बाइट्समध्ये आणि हार्डवेअरसाठी गेट्स किंवा ट्रान्झिस्टरमध्ये मोजली जाते.

8) परफॉर्मन्स: सिस्टमचा अंमलबजावणी वेळ किंवा थ्रूपुट.

9)पॉवर कंन्जमशन: सिस्टमद्वारे वापरल्या जाणाऱ्या वीजेचे प्रमाण, जे बॅटरीचे आयुष्यमान किंवा आयसीच्या कूलिंग आवश्यकता ठरवते, कारण अधिक शक्ती म्हणजे अधिक उष्णता.

10) फ्लेक्झिबिलिटी: जास्त एनआरई खर्च न घेता सिस्टमची कार्यक्षमता बदलण्याची क्षमता. सॉफ्टवेअर सामान्यतः खूप लवचिक मानले जाते.

11)मार्केटिंगसाठी लागणारा वेळ (Time to Market): सिस्टम ग्राहकांना विकता येईल इतक्या प्रमाणात सिस्टम डिझाइन आणि उत्पादन करण्यासाठी लागणारा वेळ. सिस्टम बिघाड दरम्यान सिस्टमने इतरांना हानी पोहोचवू नये.

12) प्रोटोटाइप करण्यासाठी लागणारा वेळ: सिस्टमची वर्किंग व्हर्जनतयार करण्यासाठी लागणारा वेळ, जी अंतिम सिस्टम अंमलबजावणीपेक्षा मोठी किंवा जास्त महाग असू शकते, परंतु सिस्टमची युजफुलनेस आणि करेक्टनेस रिफाईनकरण्यासाठी आणि सिस्टमची कार्यक्षमता सुधारण्यासाठी वापरली जाऊ शकते.

13) मेन्टेनॅबिलिटी (Maintainability): देखभाल क्षमता हे सिस्टीमचे सर्वात महत्त्वाचे वैशिष्ट्य आहे ज्यामध्ये ठराविक वेळेच्या अंतराने सिस्टीम दुरुस्ती किंवा बदलली जाऊ शकते.

14)अचुकता (correctness): हे उत्पादन योग्य असल्याचा निष्कर्ष काढण्यासाठी चाचणी सर्किटरी वापरून डिझाईन करण्याच्या संपूर्ण प्रक्रियेदरम्यान प्रणालीची क्षमता कार्यक्षमता तपासून कार्यान्वित प्रणालीची कार्यक्षमता योग्य असल्याची सूचित करते.

15)सुरक्षितता (safety): सिस्टीम बिघाडाच्या वेळी आणि होणार नाही याची काळजी घेतली जाते.

○

1.3 एम्बेडेड सिस्टमचे वर्गीकरण

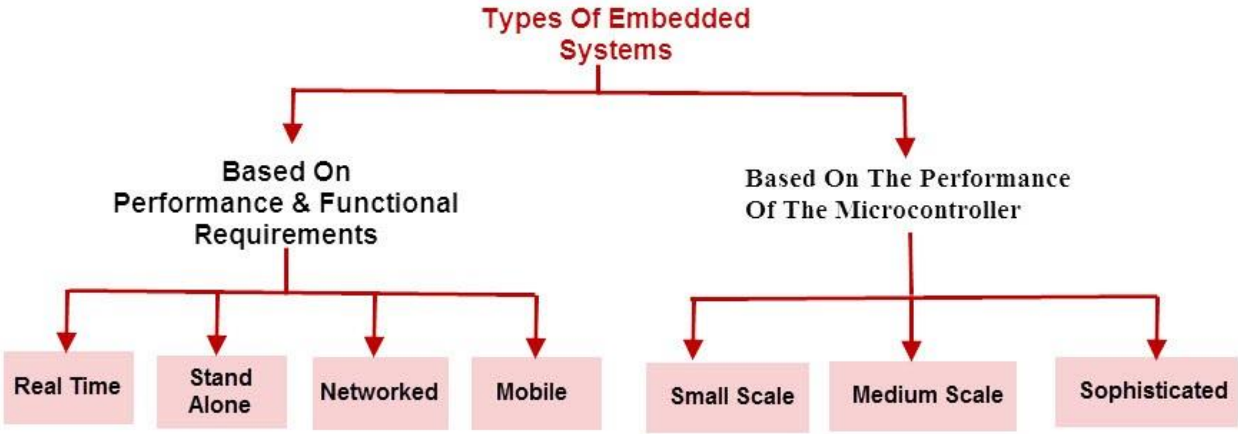


Figure1.3.1 एम्बेडेड सिस्टमचे वर्गीकरण

स्मॉल स्केल एम्बेडेड सिस्टीम:

या प्रकारच्या एम्बेडेड सिस्टीम एकाच 8-बिट किंवा 16-बिट मायक्रोकंट्रोलरने डिझाइन केल्या जातात. त्यांच्याकडे लहान आकाराचे हार्डवेअर, सॉफ्टवेअर कॉम्प्लेक्सिटी असतात आणि त्यामध्ये बोर्ड-लेव्हल डिझाइनचा समावेश असतो. ते बॅटरीवर देखील चालतात. जेव्हा या लहान आकाराच्या हार्डवेअरसाठी एम्बेडेड सॉफ्टवेअर विकसित केले जात असते, तेव्हा वापरल्या जाणाऱ्या मायक्रोकंट्रोलर किंवा प्रोसेसरसाठी विशिष्ट एडिटर, असेंबलर किंवा क्रॉस असेंबलर हे मुख्य प्रोग्रामिंग टूल्स असतात. सामान्यतः, या सिस्टीम विकसित करण्यासाठी 'सी प्रोग्रामिंग लँग्वेज' वापरली जाते. 'सी' प्रोग्राम लँग्वेजअसेंब्लीमध्ये केले जाते आणि एक्झिक्युटेबल कोड नंतर सिस्टम मेमरीमध्ये योग्यरित्या स्थित केले जातात. सॉफ्टवेअर विद्यमान मेमरीमध्ये बसले पाहिजे आणि सिस्टम सतत चालू असताना पॉवर डिसिपेशन मर्यादित करण्याची आवश्यकता लक्षात ठेवावी.

मीडियम स्केलएम्बेडेड सिस्टीम:

या सिस्टीम सहसा एक किंवा काही 16-बिट किंवा 32-बिट मायक्रोकंट्रोलर किंवा डिजिटल सिग्नल प्रोसेसर (DSPs) किंवा रिड्युस्ड इंस्ट्रक्शन सेट कॉम्प्युटर (RISCs) वापरून डिझाइन केल्या जातात. या सिस्टीममध्ये हार्डवेअर आणि सॉफ्टवेअर दोन्ही कॉम्प्लेक्सिटी असतात. मध्यम प्रमाणात एम्बेडेड सिस्टमच्या जटिल सॉफ्टवेअर डिझाइनसाठी, खालील प्रोग्रामिंग टूल्स आहेत: RTOS, सोर्स कोड इंजिनिअरिंग टूल, सिम्युलेटर, डीबगर आणि इंटिग्रेटेड डेव्हलपमेंट एन्व्हायर्नमेंट (IDE). कॉम्प्लेक्स सिस्टिममध्ये सॉफ्टवेअर टूल्स हार्डवेअरला क्लीरिफिकेशन देतात. प्रोग्रामिंग टूल म्हणूनअसेंबलरचा जास्त वापर होत नाही. या सिस्टम्स विविध फंक्शन्ससाठी सहज उपलब्ध असलेल्या ॲप्लिकेशन-स्पेसिफिक स्टँडर्ड प्रोडक्ट (ASSPs) आणि IPs चा वापर देखील करू शकतात. उदाहरणार्थ, बस इंटरफेसिंग, एन्क्रिप्टिंग, डिक्सीफरिंग, डिस्क्रिट कोसाइन ट्रान्सफॉर्मेशन आणि इनव्हर्स ट्रान्सफॉर्मेशनसाठी, TCP/IP प्रोटोकॉल स्टॅकिंग आणि नेटवर्क कनेक्टिंग फंक्शन्स आहे.

सोफिस्टिकेटेड एम्बेडेड सिस्टम्स:

सोफिस्टिकेटेड एम्बेडेड सिस्टम्समध्ये मोठ्या प्रमाणात हार्डवेअर आणि सॉफ्टवेअर कॉम्प्लेक्सिटी असतात आणि त्यांना ASIPs, IPs आणि PLAs स्केलेबल किंवा कॉन्फिगर करण्यायोग्य प्रोसेसर आणि प्रोग्रामेबल लॉजिक ॲरेची आवश्यकता असू शकते.ते अत्याधुनिक ॲप्लिकेशन्ससाठी वापरले जातात ज्यांना अंतिम सिस्टममध्ये हार्डवेअर आणि सॉफ्टवेअर सह-डिझाइन आणि एकत्रीकरण आवश्यक असते.त्यांच्या हार्डवेअर युनिट्समध्ये उपलब्ध असलेल्या प्रोसेसिंग स्पीडमुळे ते मर्यादित आहेत.वेळ वाचवून अतिरिक्त गती मिळविण्यासाठी हार्डवेअरमध्ये एन्क्रिप्शन आणि डिक्सीफरिंगअल्गोरिदम, डिस्क्रिट कोसाइन ट्रान्सफॉर्मेशन आणि इनव्हर्स ट्रान्सफॉर्मेशन अल्गोरिदम, TCP/IP प्रोटोकॉल स्टॅकिंग आणि नेटवर्क ड्रायव्हर फंक्शन्स यासारखी काही सॉफ्टवेअर फंक्शन्स अंमलात आणली जातात.सिस्टममधील हार्डवेअर रिसोर्सेसची काही फंक्शन्स देखील सॉफ्टवेअरद्वारे अंमलात आणली जातात.या सिस्टीमसाठी डेव्हलपमेंट टूल्स रिझनेबल किमतीत सहज

उपलब्ध नसतील किंवा अजिबात उपलब्ध नसतील.काही केसेस मध्ये यासाठी कंपाइलर किंवा रीटार्गेटेबल (कंपाइलर विशिष्ट टारगेट कॉन्फिगर करतो) कंपाइलर विकसित करावे लागू शकते.

स्टँडअलोन एम्बेडेड सिस्टम्स:

स्टँडअलोन एम्बेडेड सिस्टम्सना संगणकासारख्या होस्ट सिस्टमची आवश्यकता नसते, ते स्वतःच काम करते. ते ॲनालॉग किंवा डिजिटल इनपुट पोर्टमधून इनपुट घेते आणि डेटा प्रक्रिया करते, कॅल्क्युलेट करते आणि कन्वर्टिंग करते आणि रिझल्टिंग डेटा कनेक्ट केलेल्या डिव्हाइसद्वारे देते, जे कनेक्ट केलेल्या डिव्हाइसेसना नियंत्रित करते, ड्राईव्ज अँड डिस्प्लेकरते. स्टँडअलोन एम्बेडेड सिस्टम्सची उदाहरणे म्हणजे mp3 प्लेअर्स, डिजिटल कॅमेरे, व्हिडिओ गेम कन्सोल, मायक्रोवेव्ह ओव्हन आणि तापमान मापन सिस्टम.

रिअल टाइम/रिॲक्टिव्ह एम्बेडेड सिस्टम:

रिअल टाइम एम्बेडेड सिस्टम म्हणजे अशी सिस्टम जी विशिष्ट वेळेत आवश्यक आउटपुट देते. या प्रकारच्या एम्बेडेड सिस्टम कार्य पूर्ण करण्यासाठी वेळेच्या अंतिम मुदतींचे पालन करतात. जेव्हा एखाद्या सिस्टमला एखाद्या घटनेला किंवा विनंतीला काटेकोरपणे डिफाइनडटाईममध्ये प्रतिसाद द्यावा लागतो, तेव्हा आपण त्याला रिअल-टाइम सिस्टम म्हणतो. या डिफाइनड वेळेला डेडलाईनम्हणून संबोधले जाते. या सिस्टम्स प्रेडिक्टेबल असल्या पाहिजेत आणि म्हणूनच त्या डेडलाईनुसार डीटरमिनॅस्टीकअसतात.

रिअल टाइम एम्बेडेड सिस्टमचे प्रकार:

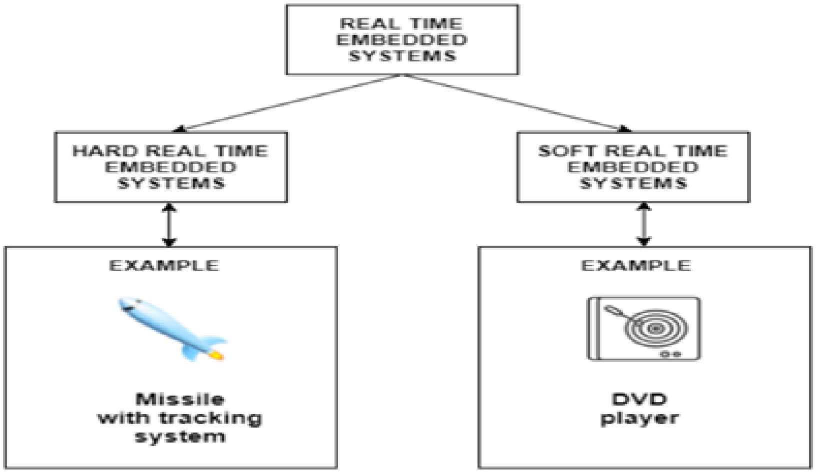


Figure 1.3.2 रिअल टाइम एम्बेडेड सिस्टमचे प्रकार

हार्ड रिअल टाइम एम्बेडेड सिस्टम:

एम्बेडेड सिस्टम जिथे रिस्पॉन्स टाइम मर्यादा पूर्ण करण्यात अपयश येते म्हणजेच डेडलाइनमुळे मिसाईल सिस्टम फेलल्युलरहोतो त्यांना हार्ड रिअल टाइम एम्बेडेड सिस्टम म्हणतात. हार्ड रिअल टाइमसाठी वेळ मर्यादा मायक्रो सेकंद आणि काही मिली सेकंद दरम्यान असते.

उदा. बटण दाबल्यानंतर निर्दिष्ट वेळेत क्षेपणास्त्र प्रक्षेपित करण्याची अंतिम मुदत चुकवल्याने लक्ष्य चुकू शकते, ज्यामुळे आपत्ती येते. हार्ड रिअल टाइम सिस्टम क्षेपणास्त्रे, विमाने, कार स्किडिंग, अँटीलॉक ब्रेक इत्यादी विविध क्षेत्रांमध्ये वापरली जातात.

सॉफ्ट रिअल टाइम एम्बेडेड सिस्टम:

सॉफ्ट रिअल टाइम म्हणजे ज्यामध्ये परफॉर्मन्सडिग्रेडेड होते परंतु प्रतिसाद वेळेच्या मर्यादा पूर्ण करण्यात फेल झाले तरी डिस्ट्रॉय होत नाही. डेडलाइन पूर्ण न झाल्यासही सिस्टम कार्य करते. वेळ मर्यादा सेकंदाच्या अंश आणि काही सेकंदांच्या दरम्यान असते.

उदा. मोबाइल फोन, डिजिटल कॅमेरा, डीव्हीडी प्लेयर इ.

1.4. वॉन न्यूमन आणि हार्वर्ड आर्किटेक्चर:

1.4.1 वॉन न्यूमन आर्किटेक्चर:

वॉन-न्यूमन आर्किटेक्चर इन्स्ट्रक्शन आणि डेटासाठी सिंगल मेमरी वापरते म्हणजेच इन्स्ट्रक्शन आणि डेटासाठी शेअर्ड मेमरी. त्यात सिंगल ॲड्रेस आणि डेटा बस आहे. सूचना आणि डेटा शेअर्ड मेमरीमध्ये असल्याने ते अनुक्रमिक क्रमाने आणावे लागतात (ते एकाच वेळी आणता येत नाहीत), त्यामुळे ते ऑपरेशनल बँडविड्थ मर्यादित करते. वॉन-न्यूमन आर्किटेक्चर इन्स्ट्रक्शनची डिझाईनईजी आहे. हे बहुतेक बाह्य मेमरी इंटरफेस करण्यासाठी वापरले जाते.

उदा. - 8085, 8086 मायक्रोप्रोसेसर

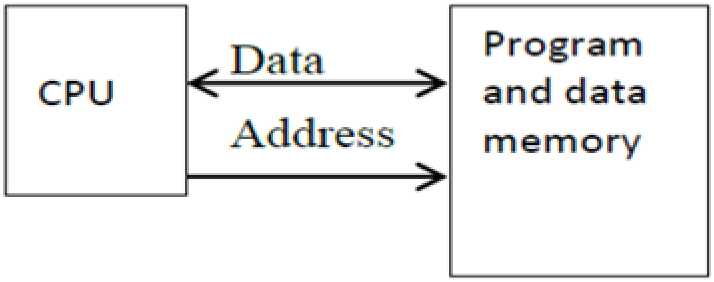


Figure.1.4.1 वॉन न्यूमन आर्किटेक्चर

1.4.2 हार्वर्ड आर्किटेक्चर:

हार्वर्ड आर्किटेक्चर प्रोग्राम आणि डेटासाठी फिजिकली स्वतंत्र मेमरी वापरते. त्यामध्ये प्रोग्राम आणि डेटासाठी स्वतंत्र बसेस आहेत. त्याची रचना कॉम्प्लेक्सआहे.

परंतु इन्स्ट्रक्शन आणि डेटा एकाच वेळी मिळवता येतो कारण इन्स्ट्रक्शन आणि डेटासाठी स्वतंत्र बसेस आहेत ज्यामुळे त्याची ऑपरेशनल बँडविड्थ वाढते.

उदा. 8051 मायक्रोकंट्रोलर

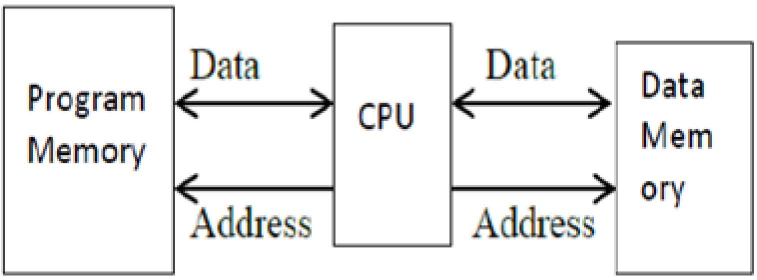
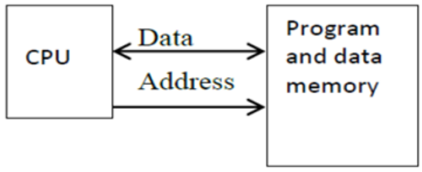
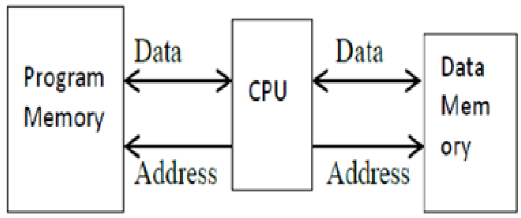


Figure 1.4.2 हार्वर्ड आर्किटेक्चर

हार्वर्ड आणि वॉन न्यूमन आर्किटेक्चरमधील फरक:

Table 1.4.3 हार्वर्ड आणि वॉन न्यूमन आर्किटेक्चरमधील तुलना

पॅरामीटर्स	हार्वर्ड आर्किटेक्चर	वॉन न्यूमन आर्किटेक्चर

FIGURE		
मेमरी सेपरेशन	प्रोग्राम (इन्स्ट्रक्शन) आणि डेटासाठी वेगळी मेमरी.	प्रोग्राम आणि डेटा दोन्हीसाठी एकच सामायिक मेमरी.
बसची रचना	इन्स्ट्रक्शन आणि डेटासाठी स्वतंत्र बसेस	इन्स्ट्रक्शन आणि डेटा ट्रान्सफर दोन्हीसाठी एकच बस.
स्पीड	इन्स्ट्रक्शन आणि डेटा एकाच वेळी मिळवता येत असल्याने एक्झिक्युशन जलद होते.	शेअर्ड बसमुळे अडथळा निर्माण होत असल्याने एक्झिक्युशनस्लो होते.
कॉम्प्लेक्स सिटी	ड्युअल मेमरी आणि बसेसमुळे अधिक जटिल हार्डवेअर.	सोपी हार्डवेअर डिझाइन.
कॉस्ट	अतिरिक्त मेमरी आणि बस हार्डवेअरमुळे जास्तखर्चिक.	शेअर्ड मेमरी आणि बसमुळे कमी खर्चिक.
प्रोसेसरएक्साम्पल्स	डीएसपी (डिजिटल सिग्नल प्रोसेसर) आणि काही मायक्रोकंट्रोलर्समध्ये (उदा. पीआयसी, एव्हीआर) वापरले जाते.	x86, ARM कॉर्टेक्स-A सिरीज सारख्या सामान्य-उद्देशीय प्रोसेसरमध्ये वापरले जाते.
पॅरललिझम	पॅरललिझमइन्स्ट्रक्शन आणि डेटा आणण्यास समर्थन देते.	एकाच बस मुळेसिक्वेन्शिअलएक्झिक्युशन.

1.5 8051 मायक्रोकंट्रोलरची वैशिष्ट्ये:

1. 8-बिट डेटा बस आणि 8-बिट ALU.
2. 16-बिट ॲड्रेस बस - जास्तीत जास्त 64KB RAM आणि ROM ॲक्सेस करू शकते.
3. ऑन-चिप RAM-128 बाइट्स (डेटा मेमरी)
4. ऑन-चिप ROM - 4 KB (प्रोग्राम मेमरी)
5. चार 8-बिट बाय-डायरेक्शनल इनपुट/आउटपुट पोर्ट चार ८-बिट बाय-डायरेक्शनल इनपुट/आउटपुट पोर्ट.
6. प्रोग्रामेबल सिरीयल पोर्ट म्हणजे एक UART (सिरीयल पोर्ट)
7. दोन 16-बिट टाइमर- टाइमर 0 आणि टाइमर 1
8. 11.0592MHz च्या क्रिस्टल फ्रिक्वेन्सीवर काम करते
9. कोणतेही ऑपरेशन न केल्यास मायक्रोकंट्रोलरमध्ये पॉवर सेव्हिंग आणि आयडल मोड असतो.
10. पाच इंटरप्ट्स उपलब्ध आहेत: दोन इंटरप्ट्स टायमर म्हणजे टायमर 0 आणि टायमर 1, दोन बाह्य हार्डवेअर इंटरप्ट्स- INT0 आणि INT1, रिसीव्ह आणि ट्रान्समिट दोन्हीसाठी सिरीयल कम्युनिकेशन इंटरप्ट (TI/RI).

1.5.1.8051चे पिन कॉन्फिगरेशन

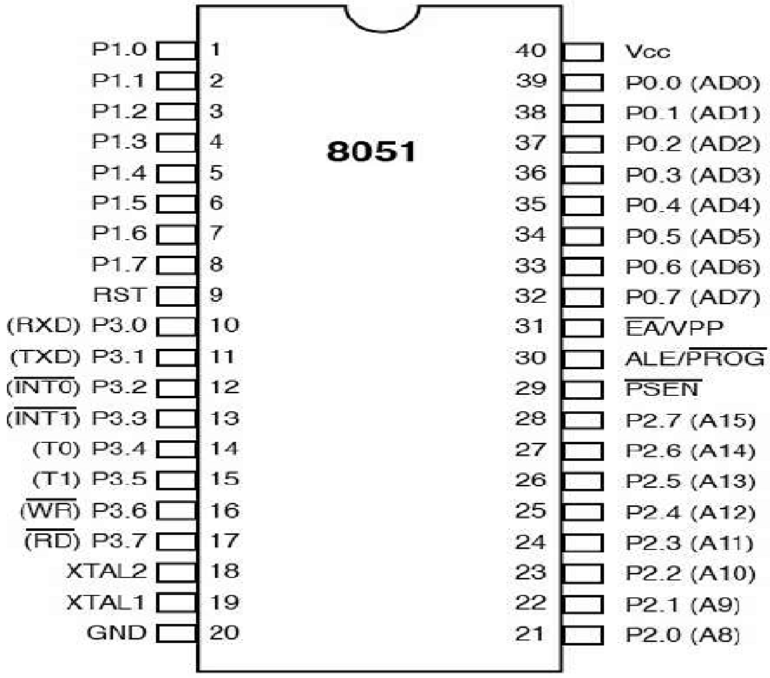


Figure1.5.1 8051 चे पिन कॉन्फिगरेशन

पिन 1 ते पिन 8 (पोर्ट 1) - पिन 1 ते पिन 8 हे साध्या I/O ऑपरेशन्ससाठी पोर्ट 1 ला असाइन केले आहेत. लॉजिक कंट्रोलवर अवलंबून ते इनपुट किंवा आउटपुट पिन म्हणून कॉन्फिगर केले जाऊ शकतात म्हणजेच जर लॉजिक झिरो (0) I/O पोर्टवर लागू केले तर ते आउटपुट पिन म्हणून काम करेल आणि जर लॉजिक वन (1) लागू केले तर पिन इनपुट पिन म्हणून काम करेल. या पिनना P1.0 ते P1.7 असेही संबोधले जाते (जिथे P1 दर्शवितो की तो पोर्ट 1मधील पिन आहे आणि '.' नंतरचा नंबर पिन नंबर दर्शवितो म्हणजेच ० पोर्टचा पहिला पिन दर्शवितो. म्हणून, P1.0 म्हणजे पोर्ट 1 चा पहिला पिन, P1.1 म्हणजे पोर्ट 1 चा दुसरा पिन आणि असेच). हे पिन द्विदिशात्मक पिन आहेत.

पिन 9 (RST) - पिन रीसेट. हा एक ॲक्टिव्ह हाय, इनपुट पिन आहे. म्हणून जर किमान 2 मशीन सायकलसाठी RST पिन हाय असेल, तर मायक्रोकंट्रोलर रीसेट होईल म्हणजेच तो सर्व ॲक्टिव्हिटीज टर्मिनेटकरेल. याला अनेकदा "पॉवर-ऑन-रीसेट" पिन असे संबोधले जाते कारण पॉवर चालू असताना (उच्च) मायक्रोकंट्रोलरला त्याच्या सुरुवातीच्या मूल्यांवर रीसेट करण्यासाठी वापरला जातो.

पिन 10 ते पिन 17 (पोर्ट 3) - पिन 10 ते पिन 17हे पोर्ट ३ पिन आहेत ज्यांना P3.0 ते P3.7 असेही म्हटले जाते. हे पिन पोर्ट 1 सारखेच आहेत आणि ते युनिव्हर्सल इनपुट किंवा आउटपुट पिन म्हणून वापरले जाऊ शकतात. हे पिन बाय डिरेकशनल पिन आहेत. या पिनमध्ये काही अतिरिक्त कार्ये देखील आहेत जी खालीलप्रमाणे आहेत:

P3.0 (RXD):10 वा पिन RXD (सिरियल डेटा रिसीव्ह पिन) आहे जो सिरियल इनपुटसाठी आहे. या इनपुट सिग्नलद्वारे मायक्रोकंट्रोलर सिरियल कम्युनिकेशनसाठी डेटा प्राप्त करतो.

P3.1 (TXD):11 वा पिन TXD (सिरियल डेटा ट्रान्समिट पिन) आहे जो सिरियल आउटपुट पिन आहे. या आउटपुट सिग्नलद्वारे मायक्रोकंट्रोलर सिरीयल कम्युनिकेशनसाठी डेटा ट्रान्समिट करतो.

P3.2 आणि P3.3 (INT0’, INT1’): 12 वी आणि 13 वी पिन अनुक्रमे एक्सटर्नल हार्डवेअर इंटरप्ट 0 आणि इंटरप्ट 1 साठी आहेत. जेव्हा हा इंटरप्ट सक्रिय केला जातो (म्हणजे जेव्हा तो कमी असतो), तेव्हा 8051 जे काही करत आहे त्यात व्यत्यय येतो आणि इंटरप्टच्या वेक्टर व्हॅल्यूवर जातो (INT0 साठी 0003H आणि INT1 साठी 0013H) आणि त्या वेक्टर जंम्प करून इंटरप्ट सर्व्हिस रूटीन (ISR) सुरू करतो.

P3.4 आणि P3.5 (T0 आणि T1): 14 वी आणि 15वी पिन टाइमर 0 आणि टाइमर 1 बाह्य इनपुटसाठी आहेत. त्यांना 16-बिट टाइमर/काउंटरने कनेक्ट केले जाऊ शकते.

P3.6 (WR’):16 वी पिन एक्सटर्नल मेमरी राइटसाठी आहे म्हणजेच एक्सटर्नल मेमरीमध्ये डेटा लिहिण्यासाठी आहे.

P3.7 (RD’):17वी पिन एक्स्टर्नल मेमरी रिडकरण्यासाठीवापरतात.

पिन 18 आणि पिन 19 (XTAL2 आणि XTAL1) – हे पिन एक्स्टर्नल ऑसिलेटरशी जोडलेले आहेत जे सामान्यतः क्वार्ट्ज क्रिस्टल ऑसिलेटर असते. ते 4MHz ते 30MHz पर्यंत एक्स्टर्नल क्लॉक
फ्रिक्वेन्सी प्रोव्हाइडकरण्यासाठी वापरले जातात.

पिन 20 (GND) – हा पिन ग्राउंड जोडलेला आहे. त्याला 0V पॉवर सप्लाय द्यावा लागतो. म्हणून तो पॉवर सप्लायच्या नकारात्मक टर्मिनलशी जोडलेला आहे.

पिन 21 ते पिन 28 (पोर्ट 2) – पिन 21 ते पिन 28 हे पोर्ट 2 पिन आहेत ज्यांना P2.0 ते P2.7 असेही म्हणतात. जेव्हा अतिरिक्त बाह्य मेमरी 8051 मायक्रोकंट्रोलरशी जोडली जाते, तेव्हा पोर्ट 2 चे पिन हायर ऑर्डर ऍड्रेसबाइट्स म्हणून काम करतात. हे पिन बाय डिरेकशनलअसतात.

पिन 29 (PSEN) – PSEN म्हणजे प्रोग्राम स्टोअर सक्षम. हे आउटपुट, सक्रिय-कमी पिन आहे. हे एक्स्टर्नल मेमरी वाचण्यासाठी वापरले जाते. 8051 आधारित सिस्टीममध्ये जिथे एक्स्टर्नल ROM प्रोग्राम कोड ठेवतो, तिथे हा पिन ROM च्या OE पिनशी जोडलेला असतो.

पिन 30 (ALE/ PROG) – ALE म्हणजे ॲड्रेस लॅच एनेबल. हा इनपुट, ऍक्टिव्हहाय पिन आहे. जेव्हा अनेक मेमरी चिप्स वापरल्या जातात तेव्हा मेमरी चिप्समध्ये फरक करण्यासाठी हा पिन वापरला जातो. पोर्ट ० वर उपलब्ध असलेले मल्टीप्लेक्स्ड ॲड्रेस आणि डेटा सिग्नल डी-मल्टीप्लेक्स करण्यासाठी देखील याचा वापर केला जातो. फ्लॅश प्रोग्रामिंग दरम्यान म्हणजेच EPROM च्या प्रोग्रामिंग दरम्यान, हा पिन प्रोग्राम पल्स इनपुट (PROG) म्हणून काम करतो.

पिन 31 (EA/ VPP) – EA म्हणजे एक्सटर्नल ॲक्सेस इनपुट. एक्स्टर्नलमेमरी इंटरफेसिंग एनेबल /डिझाबलकरण्यासाठी याचा वापर केला जातो. 8051 मध्ये, EA Vcc शी जोडलेला असतो कारण तो प्रोग्राम साठवण्यासाठी ऑन-चिप ROM सोबत येतो. 8031 आणि 8032 सारख्या फॅमिलीतील इतर मेंबर साठीज्यामध्ये ऑन-चिप ROM नाही, EA पिन GND शी जोडलेला असतो.

पिन 32 ते पिन 39 (पोर्ट 0) - पिन 32 ते पिन 39 हे पोर्ट 0 पिन आहेत ज्यांना P0.0 ते P0.7 असेही म्हणतात. ते बाय डिरेकशनलइनपुट/आउटपुट पिन आहेत. त्यांच्याकडे कोणतेही अंतर्गत पुल-अप नाहीत. म्हणून, 10kपुल-अप रजिस्टर बाह्य पुल-अप म्हणून वापरले जातात. पोर्ट 0 ला AD0-AD7 असेही म्हटले जाते कारण 8051मल्टीप्लेक्स पिन सेव्ह करण्यासाठी पोर्ट 0 द्वारे ऍड्रेस आणि डेटा देतात.

पिन 40 (VCC) - हा पिन सर्किटला पॉवर सप्लाय व्होल्टेज म्हणजेच +5 व्होल्ट प्रदान करतो.

1.6 बुलियन प्रोसेसर

बुलियन प्रोसेसर हा 8051 मधील इंटिग्रेटेड इन्स्ट्रक्शन सेट ॲक्युमलेटर प्रोसेसर आहे.
बुलियन प्रोसेसरमध्ये खालीलआहे:

- त्याचा स्वतःचा इन्स्ट्रक्शन सेटआहे.
- त्याचा ॲक्युमलेटर कॅरी फ्लॅगआहे.
- बिट ॲड्रेस करण्यायोग्य रॅमआहे.
- बिट ॲड्रेस करण्यायोग्य I/Oआहे.

8051-मायक्रोकंट्रोलर बिट ॲड्रेसेबल मेमरी आणिबुलियन प्रोसेसरसाठी बिट मॅनिपुलेशनला सपोर्ट करतो.
8051 बुलियन प्रोसेसर बिट मॅनिपुलेशनसाठी थेट सपोर्ट देतो.
8051 प्रोसेसर हा एक सीपीयू आहे जो डेटावर काही ऑपरेशन करू शकतो आणि आउटपुट देतो.
8051 प्रोसेसरमध्ये सिंगल-बिट ऑपरेशन्ससाठी संपूर्ण बुलियन प्रोसेसर असतो.
अंतर्गत रॅममध्ये 128 ॲड्रेसेबल बिट्स असतात आणि एसएफआर स्पेस 128 इतर ॲड्रेसेबल बिट्सना सपोर्ट करते. सर्व पोर्ट लाईन्स बिट-ॲड्रेसेबल असतात.

1.6.1 पॉवर सेव्हिंग मोड्स:

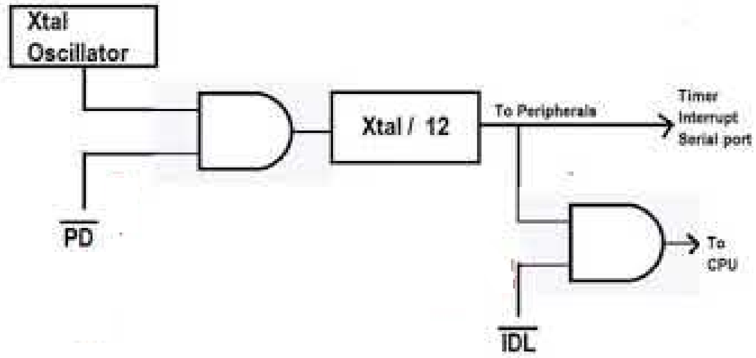


Figure1.6.1 पॉवर सेव्हिंग मोड्स

आयडियल मोड:

आयडियल मोड मध्येअंतर्गत क्लॉकसिग्नल CPU ला गेट केला जातो, परंतु इंटरप्ट, टाइमर आणि सिरीयल पोर्ट फंक्शन्सना नाही. CPU स्थिती पूर्णपणे जतन केली जाते, स्टॅक पॉइंटर, प्रोग्राम काउंटर, प्रोग्राम स्टेटस वर्ड, ॲक्युमलेटर आणि इतर सर्व रजिस्टर्स आयडियल दरम्यान त्यांचा डेटा राखतात. पोर्ट पिन आयडियल मोडसक्रिय होताना त्यांची लॉजिकल स्थिती धारण करतात. ALE आणि PSEN हाय लेवल्स होल्ड करतात आयडियल मोडक्लिअर करण्याचे दोन मार्ग आहेत. i) कोणत्याही एनेबल इंटरप्टसक्रियतेमुळे PCON.O टर्मिनेटआणि आयडियल मोडसमाप्त होईल. ii) हार्डवेअर रीसेट: म्हणजेच RST पिनवरील सिग्नल थेट PCON रजिस्टरमध्ये IDEAL बिट साफ करतो. यावेळी, CPU प्रोग्राम अंमलबजावणी जिथून सोडली होती तिथून पुन्हा सुरू करतो.

पॉवर डाउन मोड:

PCON.1 सेट करणारी सूचना पॉवर डाउन मोडमध्ये जाण्यापूर्वी ती शेवटची अंमलात आणलेली सूचना ठरते. पॉवर डाउन मोडमध्ये, ऑन-चिप ऑसिलेटर थांबवला जातो. क्लॉक फ्रोजन, सर्व फंक्शन्स थांबतात, परंतु ऑन-चिप रॅम आणि स्पेशल फंक्शन रजिस्टर राखले जातात. पोर्ट पिन त्यांच्या संबंधित SFRS द्वारे धारण केलेल्या मूल्यांचे आउटपुट करतात. ALE आणि PSEN कमी धरले जातात. पॉवर डाउन मोडमधून टर्मिनेशन: या मोडमधून बाहेर पडणे म्हणजे हार्डवेअर रीसेट. रीसेट सर्व SFR डिफाइन्ड करते परंतु चिप रॅमवर बदलत नाही PCON (पॉवर कंट्रोल रजिस्टर) हे मोड सेट करण्यासाठी SFR वापरला जातो.

Exercise: -

TLO 1.1 Explain the block diagram of embedded system with its characteristics

1. Draw the Block diagram of embedded system and list the hardware components in it. (U)
2. Explain any two of embedded system. (U)
3. Explain any four blocks characteristics of embedded system. (U)
4. List characteristics of embedded system. (R)

TLO 1.2 Classify the Embedded Systems.

1. Explain any two types of embedded system. (U)
2. Classify embedded System. (U)

TLO 1.3 Distinguish between Von-Neumann and Harvard architecture.

1. Compare Von-Neumann & Harvard architecture. (any four points) (U)

TLO 1.4 Explain the features and pin configure**uration of 8051 microcontroller.**

1. List any eight features of 8051 microcontroller. (R)
2. Draw the pin diagram of 8051 microcontroller. (R)

TLO 1.5 Explain power- saving option.

1. Draw the format of PCON register. (R)
2. Explain power saving modes of 8051. (U)

युनिट 2. 8051 प्रोग्रॅमिंग युजिंग एम्बेडेड 'C'
(8051 Programming using Embedded ‘C’)

Course Outcomes: Use embedded ‘C’ language for programming 8051 microcontroller.

सिद्धांत शिक्षण निष्पत्ती (Theory Learning Outcomes)
TLO 2.1: एम्बेडेड ‘C’ मधील विविध डेटा प्रकारांची यादी तयार करा.
TLO 2.2: अरीथमेटिक , लॉजिकल आणि डेटा ट्रान्सफर ऑपरेशन्ससाठी एम्बेडेड ‘C’ प्रोग्राम विकसित करा.
TLO 2.3: टाइमर वापरून एम्बेडेड ‘C’ प्रोग्राम विकसित करा.
TLO 2.4: सिरीयल ट्रान्समिशन(Serial Transmission)आणि रिसेप्शन(Reception) साठी एम्बेडेड ‘C’ प्रोग्राम विकसित करा.
TLO 2.5: 8051 मायक्रोकंट्रोलरमधील इंटरप्ट्स स्पष्ट करा.

परिचय: एम्बेडेड C(embedded ‘C’) ही एक प्रोग्रामिंग (programming) लँग्वेज आहे जी एम्बेडेड सिस्टमच्या (embedded System)विकासासाठी वापरली जाते. एम्बेडेड C मध्ये काही ॲडिशनल डेटा प्रकार(Datatypes) आणि कीवर्ड(Keyword)असतात. या लँग्वेजमध्ये bit, SFR सारखे विशेष डेटा प्रकार आहेत, जे स्पेशल फंक्शन रजिस्टरसाठी(SFR) मेमरीमध्ये वापरले जातात. एम्बेडेड C च्या मदतीने आपण हार्डवेअर उपकरणे(Hardware Devices) जसे की सेन्सर(Sensor) आणि इनपुट-आउटपुट डिव्हाइसेस(Input Output Devices) सोबत कार्य करू शकतो. Keil Compiler, SPJ Compiler, Embedded GNU C Compiler इत्यादी विविध एम्बेडेड C कम्पायलर (Compiler) वापरले जातात.

2.1.1: एम्बेडेड C डेटा प्रकार (Embedded ‘C’ Datatypes): एम्बेडेड C प्रोग्रामिंगमध्ये खालील डेटा प्रकार सर्वाधिक वापरले जातात:

Table 2.1 एम्बेडेड C डेटा प्रकार

डेटा प्रकार	आकार (बिट्स)	मूल्य श्रेणी/वापर
unsigned char	8 बिट	0 ते 255
(signed) char	8 बिट	-128 ते +127
unsigned int	16 बिट	0 ते 65535
(signed) int	16 बिट	-32768 ते +32767
S-bit	1-बिट	फक्त SFR बिट-ॲड्रेसेबल
Bit	1-बिट	फक्त RAM बिट-ॲड्रेसेबल
S-fr	1-बिट	RAM ॲड्रेस 80H-FFH मध्येच

2.1.2: डिसिजन कंट्रोल व लूपिंग (Conditional Statements and Loops): एम्बेडेड C मध्ये कंट्रोल स्ट्रक्चर वापरून कंडिशनल स्टेटमेंट्स वापरले जातात आणि पुन्हा पुन्हा वापरला जातो. यामध्ये दोन मुख्य प्रकार आहेत.

a. डिसिजन कंट्रोल (Conditional Statements) आणि लूप्स (Loops).
   डिसिजन कंट्रोल स्टेटमेंट्स (Conditional Statements):
   i. if-else स्टेटमेंट: जर कंडिशन ट्रू असेल तर ठराविक कोड वापरला जातो, अन्यथा दुसरा कोड चालतो.
   ii. switch स्टेटमेंट: एखाद्या व्हेरिएबलच्या(Variables) वेगवेगळ्या मूल्यांवर आधारित वेगळे कोड ब्लॉक्स चालतात.

b. लूपिंग (Loops):
   i. for लूप: जर कंडिशन ट्रू असेल तर हा ब्लॉक ऑफ कोड एक्झिक्युट होईल.

ii. while लूप: जर कंडिशन ट्रू असेल कोड हा वापरला जातो.
iii. do-while लूप: कोड मिनिमम वन टाईम वापरला जातो आणि नंतर कंडिशन तपासली जाते.

1. इफ स्टेटमेंट (if Statement):

Syntax

Program

```
if (condition)
{
    // Code to be executed if the condition is true
}
```

1. if-else Statement

Syntax

```
if (condition) {
   // Code to be executed if the condition is true
} else {
   // Code to be executed if the condition is false
}
```

Example Program

Executes one block of code if the condition is true; otherwise, executes another.

```
if (temperature > 50) {
   fan_on();
}
else
{
   fan_off();
}
```

3. नेस्टेड इफ- एल्स Nested if-else :

Syntax

```
if (condition1) {
   // Code to be executed if condition1 is true
} else if (condition2) {
   // Code to be executed if condition2 is true
} else {
   // Code to be executed if none of the conditions are true
}
```

Used when there are multiple conditions to check.

```
if (temperature > 50)
 {
   fan_on();
}
 else if (temperature > 30)
{
   fan_low();
}
else
```

```
{
   fan_off();
}
```

4. switch-case Statement

```
switch (expression) {
   case constant1:
      // Code to be executed if expression matches constant1
      break;
   case constant2:
      // Code to be executed if expression matches constant2
      break;
   // Add more cases as needed
   default:
      // Code to be executed if none of the cases match
      break;
}
```

Used when multiple conditions depend on a single variable.

```
switch (sensor_value)
{
   case 1:
      LED_on();
      break;
   case 2:
      LED_blink();
      break;
   default:
      LED_off();
      break;
}
```

लुपिंग कन्स्ट्रक्ट(Looping Constructs)-

while Loop

syntax

```
while (condition) {
   // Code to be executed while the condition is true
}
```

Executes as long as the condition is true.

```
while (button_pressed()) {
   LED_toggle();
}
```

2. do-while Loop

Syntax

```
do {
   // Code to be executed
} while (condition);
```

Executes at least once and then repeats if the condition is true.

```
do {
   read_sensor();
} while (sensor_not_ready());
```

3. for Loop

syntax

```
for (initialization; condition; increment/decrement) {
   // Code to be executed
}
```

Executes a fixed number of times.

```
for (int i = 0; i < 10; i++) {
   LED_toggle();
}
```

Loop Control Statements:

1.break → Exits the loop immediately.

```
while (1) {
   if (sensor_error()) {
      break;
   }
}
```

2.continue →

Skips the rest of the current iteration and moves to the next one.

```
for (int i = 0; i < 10; i++) {
   if (i == 5) {
      continue; // Skips iteration when i is 5
   }
   LED_toggle();
}
```

2.2: 8051 मायक्रोकंट्रोलर प्रोग्रामिंग (Microcontroller Programming)

(Arithmetic आणि Logical Operations) - 8051 मायक्रोकंट्रोलर ॲरिथमेटिक आणि लॉजिकल ऑपरेशन्सला सपोर्ट करतो. यामध्ये बेरीज, वजाबाकी, गुणाकार, भागाकार, AND, OR, XOR आणि बिट शिफ्टिंग यांचा समावेश आहे.

Table 2.2.1 ॲरिथमेटिक ऑपरेशन्स (Arithmetic Operations):

ऑपरेटर	वर्णन	उदाहरण
+	बेरीज: दोन ऑपरेण्ड्स जोडतो.	C = A + B
-	वजाबाकी: दुसऱ्या ऑपरेण्डला पहिल्यापासून वजा करतो.	C = A - B
*	गुणाकार: दोन ऑपरेण्ड्स गुणाकार करतो.	C = A * B
/	भागाकार: अंकात विभागणी करतो.	C = A / B
%	मॉड्युलस: उर्वरित भाग शोधतो.	C = A % B
++	इन्क्रिमेंट: 1 ने वाढवतो (Prefix/Postfix)	A++ किंवा ++A
--	डिक्रिमेंट: 1 ने कमी करतो (Prefix/Postfix)	A-- किंवा --A

Table 2.2.2 लॉजिकल ऑपरेशन्स (Logical Operations):

क्रमांक	ऑपरेटर	बिटवाइज लॉजिकल ऑपरेटर	उदाहरण
1.	NOT	~	Y = ~A
2.	AND	&	Y = A & B
3.	OR	L	Y = A l B
4.	EXOR	^	Y = A ^ B
5.	Left Shift	<< - बिट्स डावीकडे सरकवतो.	उदाहरण: U = S << 1
6.	Right Shift	>> - बिट्स उजवीकडे सरकवतो.	उदाहरण: G = H >> 4

बिटवाइज ऑपरेशन्सचे(Bitwise operation) स्पष्टीकरण:

1. NOT (~): सर्व बिट्स इन्वर्ट करतो (1 -> 0, 0 -> 1).
2. AND (&): बिटवाइज AND ऑपरेशन. दोन्ही बिट्स 1 असतील तेव्हाच निकाल 1 होतो.
3. OR (|): बिटवाइज OR ऑपरेशन. कमीत कमी एक बिट 1 असेल तर निकाल 1 होतो.
4. EXOR (^): XOR (Exclusive OR) ऑपरेशन. दोन बिट्स वेगळे असतील तर निकाल 1 होतो, अन्यथा 0.
5. Left Shift (<<): सर्व बिट्स डावीकडे सरकवतो, रिकामे बिट्स 0 ने भरतो.
6. Right Shift (>>): सर्व बिट्स उजवीकडे सरकवतो. रिकामे बिट्स 0 किंवा साइन बिटने भरले जातात (लॉजिकल/अर्थमॅटिक शिफ्ट).

Example Programs:

1. Write & execute c language program to add two 8-bit numbers.

```
#include<reg51.h>
void main(void)
{
    unsigned char x, y, z;
    x=0x34;
    y=0xa9;
    z=x+y;
    P1=z;
}
```

2. Write & execute c language program to subtract two 8 bit numbers.

```
# include<reg51.h>
void main(void)
{
    unsigned char x, y, z;
    x=0x25;
    y=0x12;
    z=x-y;
    P1=z;
}
```

3. Write & execute c language program to multiply two 8-bit numbers.

```
#include<reg51.h>
```

```
void main(void)
{
    unsigned char x, y, z;
    x=0x04;
    y=0x02;
    z=x*y;
    P1=z;
}
```

4. Write & execute c language program to divide two 8-bit numbers.

```
#include<reg51.h>
void main(void)
{
    unsigned char x, y, z;
    x=0x23;
    y=0x05;
    z=x/y;
    P1=z;
}
```

Example: Logical Operations

```
#include <reg51.h>
void main ()
{
    unsigned char x = 0x0F, y = 0xF0, result;
    result = x & y; // AND operation
    result = x | y; // OR operation
    result = x ^ y; // XOR operation
    result = ~x;   // NOT operation
    while (1);
}
```

Example: Bitwise Shift Operations

```
#include <reg51.h>
void main ()
{
    unsigned char val = 0x08; // 00001000 in binary
    val = val << 2; // Left shift by 2 positions (00100000)
    val = val >> 1; // Right shift by 1 position (00010000)
    while (1);
}
```

Data Transfer on Input/ Output (I/O) Ports

The 8051 has four I/O ports (P0, P1, P2, and P3), which can be used for input and output operations.

Example: Read Input from a Port

```
#include <reg51.h>
void main ()
{
    unsigned char input_data;
```

```
        input_data = P2; // Read data from Port 2
        while (1);
}
```

Example: Blinking LED Using Port 1

```
#include <reg51.h>
void delay ()
{
        int i, j;
        for (i = 0; i < 1000; i++)
           for (j = 0; j < 100; j++);
}
void main ()
{
   while (1)
   {
        P1 = 0xFF; // Turn ON all LEDs (Assume LEDs are connected to P1)
        delay ();
        P1 = 0x00; // Turn OFF all LEDs
        delay ();
   }
}
```

Programming:

1. Write an 8051 “C” program to send values 00-FF to Port2.

```
#include <reg51.h>
void delay()
{
      unsigned int i, j;
      for (i = 0; i < 500; i++)
      {
          for (j = 0; j < 1000; j++)
          {
          }
      }
}
void main()
{
 unsigned char value;
P2 = 0x00;
while (1)
  {
      for (value = 0x00; value <= 0xFF; value++)
     {
      P2 = value;
     delay();
     }
```

```
  }
}
```

2. Write a ‘C’ program to transfer the data from port P0 to port P1.

Program:

```
 #include<reg51.h>
 void main (void)
{
      unsigned char X; P0=0XFF; // P0 as input port
      P1=0X00;//P1 as output port
      while(1) //do forever
      { X = P0;//read port0
      P1 = X;// output data to port1
      }
}
```

3.Write “C” language program to toggle only bit P2.1 continuously without disturbing the rest of the bits of P2.

```
#include <reg51.h>
sbit mybit=P2^1;
void main(void)
{
   while (1)
   {
    mybit=1; //turn on P2.1
    mybit=0; //turn off P2.1
   }
}
```

4.Write a ‘C’ language program to mask the upper four bits of the data given in port 0 and write the answer in port 1.

Program:

```
 #include<reg51.h>
void main()
{
 unsigned char a ,b;
P0 = 0xff; //P0 as an input port
P1 = 0x00; //P1 as an output port
while(1)
    {
      a = P0;
      b=a & 0x0F; //masking upper 4 bit
      P1=b;
    }
 }
```

2.3: टाइमर/काउंटर (Timer/Counters) -TMOD, TCON, टाइमर/काउंटर मोड्स आणि C प्रोग्रॅम्स - 8051 मायक्रोकंट्रोलरमध्ये 2 टाइमर आणि काउंटर असतात, जे क्लॉक फ्रिक्वेन्सीवर (Clock Frequency) कार्य करतात.

टाइमर/काउंटरचा उपयोग टाइम डिले जनरेशन(Time delay generation), एक्सटर्नल इव्हेंट काउंटिंग(External clock counting), इत्यादींसाठी केला जातो.

8051 क्लॉक (Clock) व्यवस्थापन: प्रत्येक टाइमरला कार्य करण्यासाठी क्लॉकची आवश्यकता असते. 8051 मध्ये बाह्य क्रिस्टल (External Crystal) हा मुख्य क्लॉक सोर्स असतो. 8051 मायक्रोकंट्रोलरची इंटरनल सर्किटरी टाइमर्ससाठी क्रिस्टल वारंवारतेच्या 1/12 व्या भागाचा क्लॉक पुरवते, याला मशीन सायकल फ्रीक्वेंसी (Machine Cycle Frequency) म्हणतात.

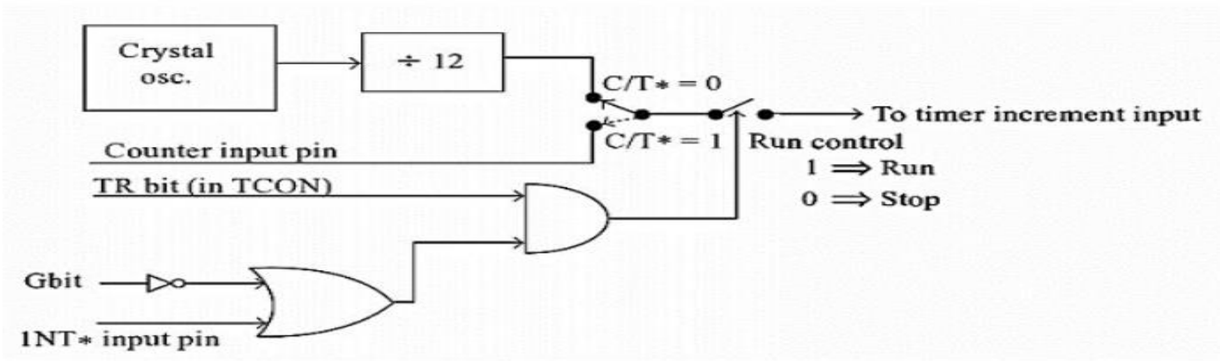


Figure 2.3 8051 टाइमर क्लॉक

उदाहरण(Example):

जर क्रिस्टल फ्रिक्वेन्सी (Crystal Frequency) 11.0592 MHz असेल, तर मायक्रोकंट्रोलर 1/12 व्या भागाचा क्लॉक पुरवेल: टाइमर क्लॉक वारंवारता = (Xtal Osc. Frequency) / 12 = (11.0592 MHz) / 12

= 921.6 KHz कालावधी (Period, T): T = 1 / (921.6 KHz)

= 1.085 मायक्रो सेकंद

2.3.1: 8051 टाइमर - 8051 मध्ये Timer0 (T0) आणि Timer1 (T1) हे 16-बिटचे दोन टाइमर्स असतात. 8051 हा 8-बिट आर्किटेक्चर असल्यामुळे, हे प्रत्येक टाइमर वेगळ्या 8-बिटच्या दोन रजिस्टरद्वारे प्रवेश करता येतात. हे रजिस्टर टाइमर काउंट लोड करण्यासाठी वापरले जातात.

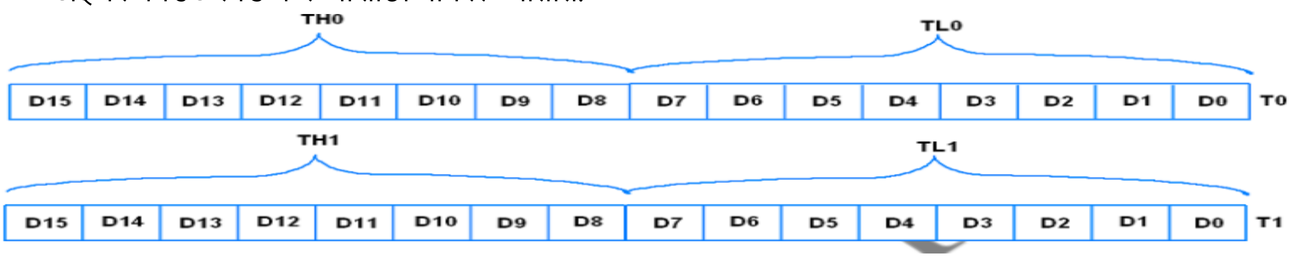


Figure.2.3.1 8051 टाइमर

8051 मध्ये टाइमर मोड रजिस्टर आणि टाइमर कंट्रोल रजिस्टर असते, जे ऑपरेशन मोड निवडण्यासाठी आणि नियंत्रणासाठी वापरले जाते.

2.3.2 TMOD हे एक 8-बिट रजिस्टर आहे, जे Timer0 आणि Timer1 साठी टाइमर मोड सेट करण्यासाठी वापरले जाते.

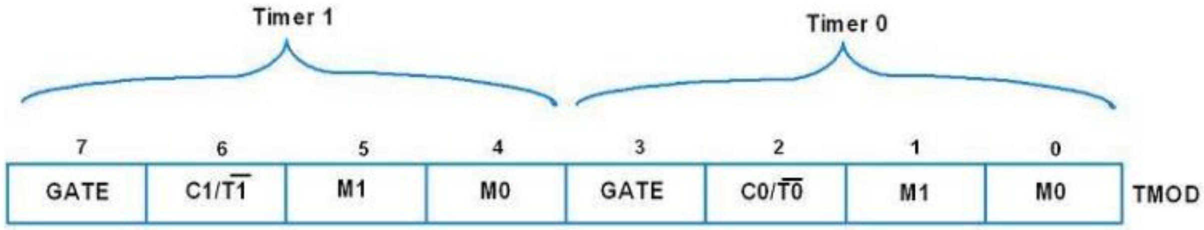


Figure.2.3.2 TMOD रजिस्टर

याच्या खालच्या 4 बिट्स Timer0 साठी आणि वरच्या 4 बिट्स Timer1 साठी वापरल्या जातात.

Bit 7,3 – GATE:

1 = केवळ INT0/INT1 पिन उच्च(High) असताना आणि TR0/TR1 सेट असताना टाइमर/काउंटर एनेबल करा.

0 = TR0/TR1 सेट असताना टाइमर/काउंटर सक्षम करा.

Bit 6,2 - C/ (Counter/Timer): टाइमर किंवा काउंटर सेलेक्ट बिट

1 = काउंटर म्हणून यूज करा

0 = टाइमर म्हणून यूज करा

Bit 5:4 & 1:0 - M1:M0: टाइमर/काउंटर मोड निवड बिट

खालील तक्त्यानुसार हे टाइमर/काउंटर मोड निवडण्यासाठी वापरले जातात.

Table 2.3.2 टाइमर/काउंटर मोड

M1	M0	Mode	Operation
0	0	0 (13-bit timer mode)	13-bit timer/counter, 8-bit of THx & 5-bit of TLx
0	1	1 (16-bit timer mode)	16-bit timer/counter, THx cascaded with TLx
1	0	2 (8-bit auto-reload mode)	8-bit timer/counter (auto-reload mode), TLx reload with the value held by THx each time TLx overflow
1	1	3 (split timer mode)	Split the 16-bit timer into two 8-bit timers i.e. THx and TLx like two 8-bit timer

2.3.3: TCON रजिस्टर

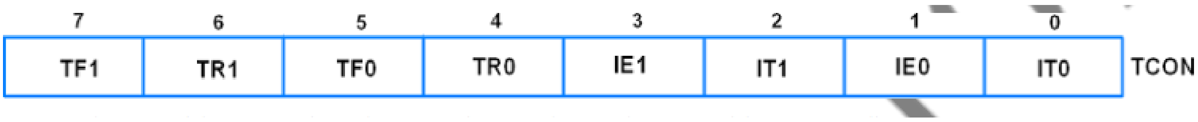


Figure.2.3.3 TCON रजिस्टर

TCON हे एक 8-बिट नियंत्रण रजिस्टर आहे आणि त्यात टाइमर आणि इंटरप्ट फ्लॅग्स (Interrupt Flag) असतात.

Bit 7 - TF1: Timer1 Overflow Flag

1 = Timer1 ओव्हरफ्लो झाला (म्हणजेच Timer1 त्याच्या कमाल मर्यादेला पोहोचून पुन्हा शून्यावर येतो).

0 = Timer1 ओव्हरफ्लो झालेला नाही.

हे सॉफ्टवेअरद्वारे क्लिअर केले जाते. Timer1 ओव्हरफ्लो इंटरप्ट सर्व्हिस रूटीन (ISR) मध्ये, ISR मधून बाहेर पडताना हे बिट आपोआप क्लिअर होईल.

Bit 6 - TR1: Timer1 Run Control Bit

1 = Timer1 सुरू.

0 = Timer1 थांबवा.

हे सॉफ्टवेअरद्वारे सेट आणि क्लिअर केले जाते.

Bit 5 – TF0: Timer0 Overflow Flag

1 = Timer0 ओव्हरफ्लो झाला (म्हणजेच Timer0 त्याच्या कमाल मर्यादेला पोहोचून पुन्हा शून्यावर येतो).

0 = Timer0 ओव्हरफ्लो झालेला नाही.

हे सॉफ्टवेअरद्वारे क्लिअर केले जाते. Timer0 ओव्हरफ्लो इंटरप्ट सर्व्हिस रूटीन (ISR) मध्ये, ISR मधून बाहेर पडताना हे बिट आपोआप क्लिअर होईल.

Bit 4 – TR0: Timer0 Run Control Bit

1 = Timer0 स्टार्ट.

0 = Timer0 स्टॉप.

Bit 3 - IE1: External Interrupt1 Edge Flag

1 = External Interrupt1 ऑकर्ड (Occurred).

0 = External Interrupt1 प्रोसेस्ड (Processed).
हे हार्डवेअरद्वारे सेट आणि क्लिअर केले जाते.

Bit 2 - IT1: External Interrupt1 Trigger Type Select Bit

1 = INT1 पिनवर फॉलिंग एज (घटणारी कडा) आल्यास इंटरप्ट होतो.
0 = INT1 पिनवरील लो लेव्हलवर इंटरप्ट होतो.

Bit 1 – IE0: External Interrupt0 Edge Flag

1 = External Interrupt0 ऑकर्ड (Occurred).
0 = External Interrupt0 प्रोसेस्ड (Processed).
हे हार्डवेअरद्वारे सेट आणि क्लिअर केले जाते.

Bit 0 – IT0: External Interrupt0 Trigger Type Select Bit

1 = INT0 पिनवर फॉलिंग एज आल्यास इंटरप्ट होतो.
0 = INT0 पिनवरील लो लेव्हलवर इंटरप्ट होतो.

8051 टाइमर मोड्स -टाइमरचे वेगवेगळे ऑपरेशन मोड असतात, जे TMOD रजिस्टरमध्ये M0 आणि M1 बिट्सच्या कॉम्बिनेशन निवडले जातात.

Mode 0 (13-बिट टाइमर मोड) -Mode 0 मध्ये 8-बिट टाइमर किंवा काउंटर 5-बिट प्री-स्केलरसह वापरला जातो. त्यामुळे तो 13-बिट टाइमर/काउंटर बनतो. या मोडमध्ये TL0 किंवा TL1 चे 5 बिट्स आणि TH0 किंवा TH1 चे सर्व 8 बिट्स वापरले जातात.

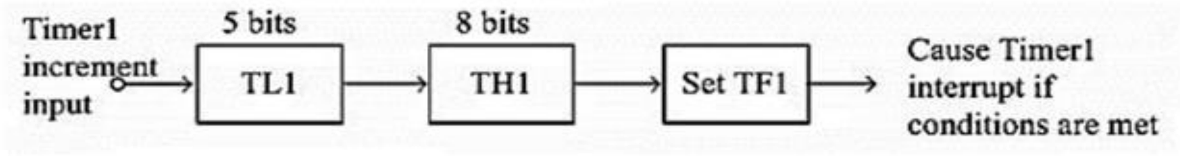


Figure.2.3.4 मोड 0

या उदाहरणात Timer1 निवडलेला आहे. या प्रकरणात प्रत्येक 32 (2^5) घटना काउंटर ऑपरेशन्ससाठी किंवा 32 मशीन सायकल्ससाठी टाइमर ऑपरेशनमध्ये, TH1 रजिस्टर 1 ने वाढवले जाते. जेव्हा TH1 FFH वरून 00H वर जाते, तेव्हा TCON रजिस्टरमधील TF1 फ्लॅग हाय होतो आणि टाइमर/काउंटर थांबतो.
उदाहरणार्थ, जर TH1 = F0H असेल आणि टाइमर मोडमध्ये असेल, तर TF1 फ्लॅग 10H * 32 = 512 मशीन सायकल्स(Machine cycle) नंतर हाय होईल.
Mode 1 (16-बिट टाइमर मोड)
Mode 1 मध्ये 16-बिट टाइमर किंवा काउंटर असतो.

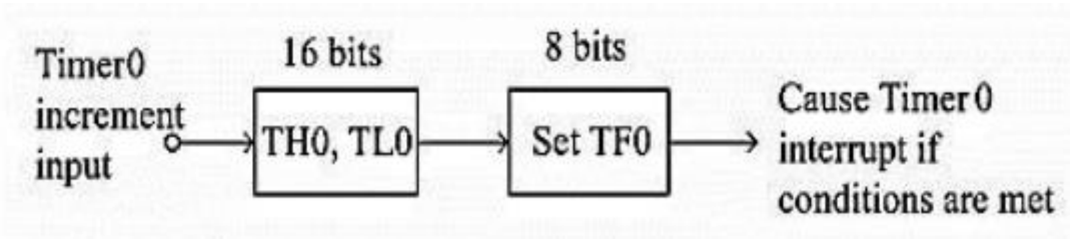


Figure.2.3.5 मोड 1

या मोडमध्ये टाइमर 0000 ते FFFFH पर्यंत मोजतो, ज्यामुळे मोठ्या प्रमाणात विलंब (delay) निर्माण करता येतो.आवश्यक विलंबासाठी TH & TL रजिस्टरमध्ये टाइमर व्हॅल्यू लोड केली जाते.टाइमर सुरू केल्यावर, तो हायर व्हॅल्यू (0xFFFF) गाठेपर्यंत मोजतो.एकदा 0xFFFF वर पोहोचल्यानंतर, तो पुन्हा शून्यावर जातो आणि ओव्हरफ्लो फ्लॅग(Overflow flag) सेट करतो. या वेळी, टाइमर व्हॅल्यू पुन्हा लोड करावी लागते आणि ओव्हरफ्लो फ्लॅग क्लिअर करावा लागतो.

Mode 2 (8-बिट ऑटो-रिलोड टाइमर/काउंटर):

Mode 2 मध्ये टाइमर 8-बिट टाइमर म्हणून वापरला जातो, जो 00 ते FFH पर्यंत मोजतो. आवश्यक डिलेसाठी TH रजिस्टरमध्ये टाइमर व्हॅल्यू लोड केली जाते, जी नंतर TL मध्ये कॉपी केली जाते. एकदा टाइमर सुरू झाल्यानंतर, तो TL वाढवतो, आणि जेव्हा तो 0xFF वर पोहोचतो, तेव्हा तो पुन्हा शून्यावर जातो, ओव्हरफ्लो फ्लॅग सेट करतो आणि TH मधून नवीन व्हॅल्यू री-लोड करतो.

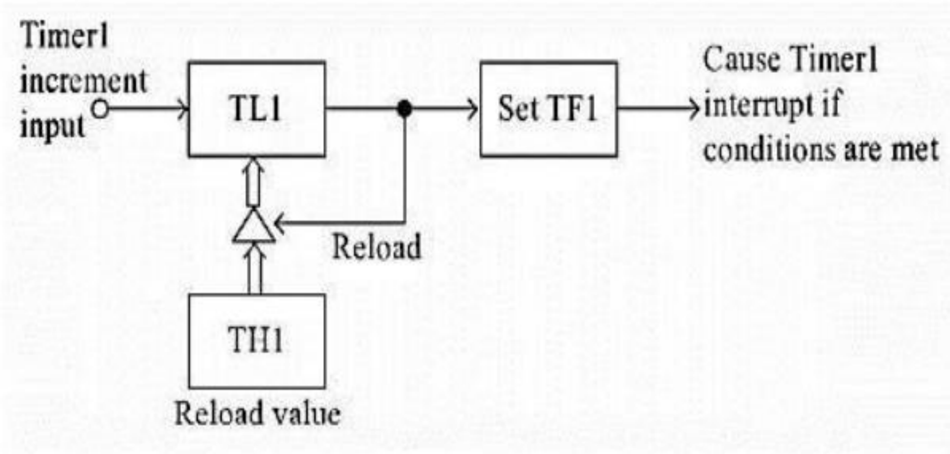


Figure.2.3.6 मोड 2

Mode 3 (स्प्लिट टाइमर मोड) -खालील आकृती मध्ये Timer0 चा Mode 3 दाखवला आहे.

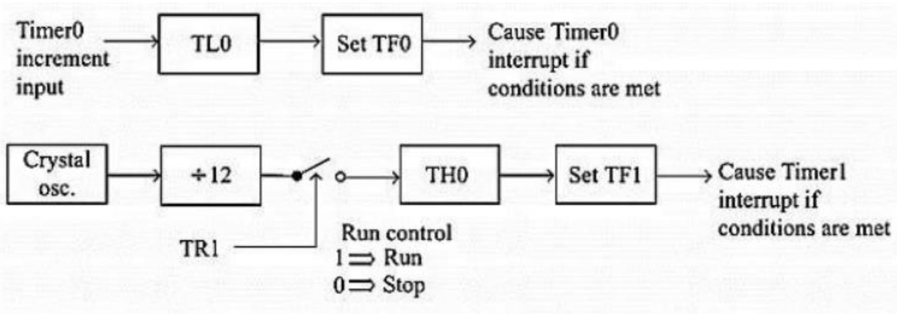


Figure.2.3.6 मोड 3

Mode 3 मध्ये Timer1 केवळ काउंट (count) ठेवतो, रन होत नाही. जर Timer0 हा Mode 3 मध्ये असेल, तर Timer1 हा Mode 0, 1 किंवा 2 मध्ये कॉन्फिगर(ConFIGUREure) केला जातो. या परिस्थितीत, Timer1 मायक्रोकंट्रोलरमध्ये इंटरप्ट (interrupt) निर्माण करू शकत नाही. Mode 3 मध्ये: जर TF1 (Timer1 Overflow Flag) वापरण्यात आला असेल, तर Timer1 चा उपयोग Baud Rate Generator म्हणून होतो. Timer0 आणि Timer1 साठी Gate बिटचे कार्य खालीलप्रमाणे आहे: TL0 च्या चालू/बंद स्थितीवर Mode 0, 1, किंवा 2 प्रमाणेच नियंत्रण ठेवते. TH0 चे नियंत्रण फक्त TR1 बिटद्वारेच होते. त्यामुळे Mode 3 मध्ये Timer0 साठी Gate बिटला कोणतेही विशिष्ट कार्य नसते. Mode 3 प्रामुख्याने अतिरिक्त 8-बिट टाइमर/काउंटर आवश्यक असलेल्या अनुप्रयोगांसाठी वापरण्यात येतो.

8051 मध्ये Mode 3 टाइमर ऑपरेशन

Mode 3 मध्ये 8051 मध्ये एकूण 3 टाइमर असतात:

TH0 → 8-बिट टाइमर

TL0 → 8-बिट टाइमर/काउंटर

Timer1 → 16-बिट टाइमर/काउंटर

जर Timer0 Mode 3 मध्ये असेल आणि Timer1 हा Mode 0, 1 किंवा 2 मध्ये कार्यरत असेल, तर:

Gate बिट लो (Low) असेल किंवा INT1 हाय (High) असेल, तेव्हा Timer1 चालू होतो.

Gate बिट हाय High) असेल आणि INT1 लो (Low) असेल, तेव्हा Timer1 बंद होतो.

8051 मध्ये टाइमर प्रोग्रॅमिंग स्टेप्स:

1. TMOD रजिस्टरमध्ये कोणता टाइमर (0 किंवा 1) आणि कोणता मोड वापरायचा ते सेट करा.
2. TL आणि TH रजिस्टरमध्ये निवडलेल्या मोडनुसार स्टार्टिंग काउंट लोड करा.
3. "TR0 = 1" (Timer0 सुरू करण्यासाठी) किंवा "TR1 = 1" (Timer1 सुरू करण्यासाठी) ही इन्स्ट्रक्शन द्या.
4. "while(TRx == 0)" च्या सहाय्याने टाइमर फ्लॅग (TF) मॉनिटर करा आणि TF = 1 झाल्यावर लूपमधून बाहेर पडा.
5. "TR0 = 0" किंवा "TR1 = 0" ने संबंधित टाइमर थांबवा.
6. "TF0 = 0" किंवा "TF1 = 0" ने पुढच्या RUNसाठी टाइमर फ्लॅग क्लिअर करा.
7. परत स्टेप 2 वर जा आणि TH व TL मध्ये नवीन व्हॅल्यू लोड करा.

1. Write C language program to generate a square wave of 2khZ on pin P1.1 by using timer 0 and mode 1. Assume crystal frequency is 11.0592MHz.

Crystal frequency= 11.0592 MHz

I/P clock = 11.0592 X 106 = 11.0592MHz

1/12x11.0592Mhz = 921.6 KHz

Tin = 1.085µ sec

For 2 kHz square wave

Fout = 2 KHz

Tout = 1/2 X 10 3

Tout = 500µ sec

Consider half of it = Tout = 250µ sec

N = Tout / Tin = 250/1.085= 230

65536-230= (65306p10) =FF1A H

Program:

```
#include<reg51.h>
void delay(void);
sbit p=P1^1;
void main (void)
{       while (1)
        { p=~p;
        delay();
        }
 }
void delay()
{
TMOD=0X01; //set timer 0 in mode 1 i.e. 16 bit timer
TL0=0X1AH; //load TL register with LSB of count
TH0=0XFFH ; //Load TH register with MSB of count
TR0 =1 //Start timer 0
While(TF0==0) //wait until timer rolls over
TR0=0; //Stop timer 0
TF0=0; //Clear timer flag 0
}
```

2. Write C program to generate a square wave on PORT 1.0 with 200uSec. time period using Timer1 in mode2. Assume crystal frequency is 11.0592MHz.

Crystal frequency= 11.0592 MHz

I/P clock = 11.0592 X 10 6 = 11.0592MHz
1/12x11.0592Mhz = 921.6 Khz
Tin = 1.085μ sec
Tsq =200μsec
Ton=Toff=200/2=100 μsec
N =Tout / Tin =100/1.085= 92.16
256-92=164=A4H

Program:

```
#include<reg51.h>
sbit sq = P1^0; /* set test pin0 of port1 */
void main()
{
TMOD = 0x20; /* Timer1 mode2 (8-bit auto reload timer mode) */
TH1 = 0xA4; /* Load 8-bit in TH1 */
TR1 = 1; /* Start timer1 */
while(1)
{
sq = ~sq; /* Toggle test pin */
while(TF1 == 0); /* Wait until timer1 flag set */
TF1 = 0; /* Clear timer1 flag */
}
}
}
```

2.4 8051 मधील सिरीयल कम्युनिकेशन (8051 Serial Communication) -89C51 हा पूर्ण-डुप्लेक्स (Full-Duplex) सिरीयल पोर्ट सपोर्ट करतो. TxD (ट्रान्समिट डेटा) आणि RxD (रिसीव्ह डेटा) पिनचा वापर अनुक्रमे डेटा पाठवण्यासाठी आणि प्राप्त करण्यासाठी होतो.सिरीयल कम्युनिकेशनसाठी दोन विशेष फंक्शन रजिस्टर (SFR) वापरली जातात.SBUF रजिस्टर (Serial Buffer Register): SBUF हे एक 8-बिट रजिस्टर आहे. ट्रान्समिशन (प्रेषण) आणि रिसेप्शन (स्वीकार) यासाठी स्वतंत्र SBUF रजिस्टर असते. TXD लाईनद्वारे डेटा ट्रान्सफर करण्यासाठी, डेटा SBUF मध्ये ठेवावा लागतो. त्याचप्रमाणे, RXD पिनद्वारे प्राप्त झालेला 8-बिट डेटा SBUF मध्ये संग्रहित केला जातो.

SCON रजिस्टर (**Serial Control Register**): SCON हे सिरीयल कंट्रोल रजिस्टर आहे, ज्यामध्ये खालील कंट्रोल बिट्स असतात: मोड सेलेक्शन साठीचे बिट्स सिरीयल पोर्ट इंटरप्ट बिट (TI आणि RI) ट्रान्समिशन व रिसेप्शनसाठीचा 9वा डेटा बिट (TB8 आणि RB8)

Figure 2.4.1 SCON register:

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0 and SM1: These are used to select the mode.

SM0	SM1	Mode	Description	Baud Rate
0	0	0	Shift Register	f_{osc}/12
0	1	1	8 bit UART	Variable (Determined by Timer 1 overflow rate)
1	0	2	9 bit UART	f_{osc}/64 or f_{osc}/32
1	1	3	9 bit UART	Variable (Determined by Timer 1

				overflow rate)

SCON रजिस्टरमध्ये असलेले बिट्स आणि त्यांचे कार्य

1. SM2 (Multiprocessor Mode Control Bit): 1 (लॉजिक 1) → मल्टीप्रोसेसर मोड सक्रिय होतो. 0 (लॉजिक 0) → सामान्य मोडमध्ये कार्य करते.
2. REN (Receiver Enable): REN = 1 → रिसीव्हर (RXD) डेटा स्वीकारेल. REN = 0 → रिसीव्हर डेटा स्वीकारणार नाही.
3. TB8 (Transmitted 9th Bit): हा 9वा बिट ट्रान्समिट करण्यासाठी वापरला जातो.
4. RB8 (Received 9th Bit): हा 9वा बिट रिसीव्ह करण्यासाठी वापरला जातो.
5. TI (Transmit Interrupt): जेव्हा SBUF मधून 8-बिट डेटा पाठवला जातो आणि SBUF रिकामे होते, तेव्हा TI = 1 होतो. TI = 1 झाल्यावर, प्रोसेसरला इंटरप्ट पाठवले जाते.
6. RI (Receive Interrupt): जेव्हा SBUF मध्ये 8-बिट डेटा प्राप्त होतो, तेव्हा RI = 1 होतो. RI = 1 झाल्यावर, प्रोसेसरला इंटरप्ट पाठवले जाते. RI = 1 होण्याआधी, SM2 वर आधारित एरर तपासली जाते.

2.4.2 8051 मध्ये सिरियल डेटा ट्रान्सफर करण्याची पद्धत:

1. TMOD रजिस्टरमध्ये 20H लोड करा, ज्यामुळे टायमर 1 मोड 1 मध्ये कार्यान्वित होईल आणि बौड रेट (Boud Rate) सेट होईल.
2. TH1 मध्ये योग्य मूल्य लोड करा जे सिरियल डेटा ट्रान्सफरसाठी आवश्यक बौड रेट सेट करेल.
3. SCON रजिस्टरमध्ये 50H लोड करा, जे सिरियल मोड 1 निर्दिष्ट करतो, ज्यामध्ये 8-बिट डेटा *स्टार्ट* आणि *स्टॉप बिट्स* सोबत फ्रेम केला जातो.
4. TR1 = 1 सेट करा जेणेकरून टायमर 1 सुरू होईल.
5. TI फ्लॅग CLR TI निर्देशाने साफ करा.
6. SBUF मध्ये ट्रान्सफर करावयाचा कॅरॅक्टर बाइट लिहा.
7. TI फ्लॅग मॉनिटर करा जेणेकरून कॅरॅक्टर पूर्णपणे ट्रान्सफर झाला आहे का ते तपासता येईल.
8. पुढील बाइट ट्रान्सफर करण्यासाठी, स्टेप 5 ला पुन्हा जा.

ही प्रक्रिया पूर्ण डुप्लेक्स सिरियल कम्युनिकेशनमध्ये 89C51/8051 साठी उपयुक्त आहे.

2.4.3 C Programs for serial communication:

1. Write a ‘C’ language program for serial communication to transfer letter ‘A’ serially at 4800 baud continuously.

Count calculation for 4800 baud

Baudrate = Fosc/ (32 * 12 * (256- count))

Now with Fosc = 11.0592Mhz,

count value for 4800 baudrate will be 4800 = 11.0592 X 10^6 / (32 * 12 * (256-count))

256-count = 11.0592 X 10^6 / (32 * 12 *4800) = 6

Count =256-6 = 250 = FA H

Program:

```
#include <reg51.h>
void main(void)
{
TMOD = 0x20; //Initialize timer 1 in mode 2
TH1 = 0xFD; //baud rate 9600
SCON = 0x50; //start serial communication ( 8bit , 1 stop bit ,REN )
```

```
TR1 = 1; //start timer 1
while(1)
  {
    SBUF='A'; // place value in buffer
    while( TI==0);
    TI=0; // clear TI
  }
}
```

2. Write a C language program to transfer the message “MSBTE” serially at 9600 baud rate ,8 data bit and 1 stop bit.Count calculation for 9600 baud:

Count calculation for 9600 baud

The final formula for baud rate is as below.

Baudrate = Fosc/ (32 * 12 * (256- count))

Now with Fosc = 11.0592Mhz,

count value for 9600 baudrate will be 9600 = 11.0592 X 106 / (32 * 12 * (256-count))

256-count = 11.0592 X 106 / (32 * 12 *4800) = 3

Count =256-3 = 253 = FD H

Program:

```
#include <reg51.h>
void main(void)
{
unsigned char i, text[ ] = “MSBTE”; //initialize array
TMOD = 0x20; //Initialize timer 1 in mode 2
TH1 = 0xFD; //baud rate 9600
SCON = 0x50; //start serial communication ( 8bit , 1 stop bit , REN )
TR1 = 1; //start timer 1
for(i=0;i<6;i++) //Read array and transmit serially till end
{
SBUF = text[i];
while(TI==0); //check interrupt
TI = 0; //clear interrupt
}
}
```

3.Write 89C51 “c‟ program to transfer the message “INDIA” serially at 9600 baud rate continuously. Use 8bit data and 1 stop bit.

Program:

```
#include <reg51.h>
void SerTx (unsigned char);
void main(void)
{
TMOD=0x20; //timer 1 in mode 2
TH1=0xFD; //set baud rate to9600
SCON=0x50; // 8bit data,1 stop bit, receiver enable
TR1=1; //start timer 1
```

```
 while (1) //repeat transmission
   {
     SerTx (‘I’); //send I serially
     SerTx (‘N’) //send N serially
     SerTx(‘D’); //send D serially
     SerTx(‘I’); //send I serailly
     SerTx (‘A’); //send A serially
   }
}
void SerTx (unsigned char x)
{
   SBUF=x;
   while(TI==0); //wait for transmission to complete
   TI=0; //clear TI for next transmission
}
```

सिरियल डेटा रिसेप्शनसाठी प्रोग्रामिंग स्टेप्स :

1. TMOD रजिस्टरमध्ये 20H लोड करा.
2. TH1 मध्ये योग्य मूल्य लोड करा जेणेकरून बौड रेट सेट होईल.
3. SCON रजिस्टरमध्ये 50H लोड करा, ज्यामुळे सिरियल मोड 1 एनेबल होईल (8-बिट डेटा स्टार्ट आणि स्टॉप बिट्ससह फ्रेम केला जातो).
4. TR1 = 1 सेट करा जेणेकरून टायमर सुरू होईल.
5. RI फ्लॅग CLR RI निर्देशाने साफ करा.
6. RI फ्लॅग मॉनिटर करा हे तपासण्यासाठी की पूर्ण कॅरॅक्टर (character) प्राप्त झाला आहे का.
7. RI = 1 झाल्यावर SBUF मध्ये प्राप्त बाइट साठवलेले असेल.
8. पुढील कॅरॅक्टर प्राप्त करण्यासाठी स्टेप 5 ला पुन्हा जा.

1. Write 89C51 ‘C’ program to receive data serially from RX pin and send the data on port 1 continuously .Assume baud rate to be 9600 and crystal frequency as 11.0592 MHz.

Program:

```
#include <reg51.h>
Void main (void)
{
unsigned char mybyte;
TMOD=0x20; //use timer 1,8 –bit auto-reload
TH1=0xFD; //9600 baud rate
SCON=0x50
TR1=1; //start timer
while(1) //repeat forever
      {
      while(RI==0); //wait to receive
      mybyte=SBUF; //save value
      P1=mybyte; //write value to port
      RI=0;
      }
```

}

8051 मायक्रोकंट्रोलरमधील इंटरप्ट्स(Microcontroller Interrupts): 8051 मायक्रोकंट्रोलरमध्ये टायमर आणि सिरियल इंटरप्ट्स अंतर्गत निर्माण होतात, तर बाह्य इंटरप्ट्स अतिरिक्त पेरिफेरल डिव्हाइसेस(Peripheral devices) किंवा स्विचेसद्वारे(switch) निर्माण होतात. बाह्य इंटरप्ट्स एज-ट्रिगर(external edge trigger interrupt) (फॉलिंग/रायझिंग एजवर ट्रिगर होणारे) किंवा लेव्हल-ट्रिगर (सिग्नल विशिष्ट लेव्हलपर्यंत सक्रिय राहणारे) असू शकतात. मायक्रोकंट्रोलर इंटरप्ट आल्यावर संबंधित ISR (इंटरप्ट सर्व्हिस रूटीन) अंमलात आणतो, ज्यामुळे विशिष्ट मेमरी लोकेशन्स इंटरप्टसाठी वापरल्या जातात. 8051 मध्ये इंटरप्ट स्पेशल फंक्शन रजिस्टर (SFRs) असतात, जसे की IE (Interrupt Enable) रजिस्टर, जे विशिष्ट इंटरप्ट्स एनेबल (Enable) किंवा डिसेबल (Disable) करण्यासाठी वापरले जाते, आणि IP (Interrupt Priority) रजिस्टर, जे इंटरप्ट्सच्या प्रायोरिटी (प्राधान्यक्रम) सेट करण्यासाठी वापरले जाते. 8051 मध्ये तीन स्तरांची इंटरप्ट प्रायोरिटी प्रणाली असते, ज्यामुळे उच्च प्राधान्यक्रम असलेल्या इंटरप्ट्सना आधी प्रक्रिया दिली जाते.

2.5 Interrupts: 8051 interrupts, IE and IP SFRs -8051 मायक्रोकंट्रोलरमध्ये टायमर आणि सिरियल इंटरप्ट्स इंटरनल निर्माण होतात, तर एक्सटर्नल इंटरप्ट्स एक्सटर्नल पेरिफेरल डिव्हाइसेस किंवा स्विचेसद्वारे निर्माण होतात. बाह्य इंटरप्ट्स दोन प्रकारचे असतात: एज-ट्रिगर (Edge-Triggered), जे फॉलिंग किंवा रायझिंग एजवर सक्रिय होतात, आणि लेव्हल-ट्रिगर (Level-Triggered), जे विशिष्ट सिग्नल लेव्हलपर्यंत सक्रिय राहतात. मायक्रोकंट्रोलर एखादा इंटरप्ट आल्यावर त्यावर प्रतिसाद देण्यासाठी ISR (Interrupt Service Routine) अंमलात आणतो, ज्यामुळे संबंधित मेमरी लोकेशन्स इंटरप्टशी जुळवले जातात. 8051 मध्ये इंटरप्ट्सचे व्यवस्थापन करण्यासाठी स्पेशल फंक्शन रजिस्टर (SFRs) असतात. IE (Interrupt Enable) रजिस्टर विशिष्ट इंटरप्ट्स एनेबल (Enable) किंवा डिसेबल (Disable) करण्यासाठी वापरले जाते, तर IP (Interrupt Priority) रजिस्टर इंटरप्ट्सच्या प्राधान्यक्रम (Priority) सेट करण्यासाठी वापरले जाते. 8051 मध्ये तीन स्तरांची इंटरप्ट प्रायोरिटी प्रणाली असते, ज्यामुळे उच्च प्राधान्यक्रम असलेल्या इंटरप्ट्सना आधी प्राधान्य दिली जाते.

Table 2.5 Interrupt Vector

Interrupt No.	Interrupt source	Vector Address	Priority
0	External Interrupt 0 (INT0)	0003H	1
1	Timer 0 interrupt	000BH	2
2	External Interrupt 1 (INT1)	0013H	3
3	Timer 1 interrupt	001BH	4
4	Serial Interrupt (TI/RI)	0023H	5

इंटरप्टसाठी वापरले जाणारे SFRs:

1. IE (Interrupt Enable) रजिस्टर
2. IP (Interrupt Priority) रजिस्टर

IE (Interrupt Enable) रजिस्टर:

IE रजिस्टर चा वापर इंटरप्ट सक्षम (Enable) आणि अक्षम (Disable) करण्यासाठी केला जातो. 8051 मायक्रोकंट्रोलर मध्ये हे रजिस्टर इंटरप्टच्या प्राधान्यक्रम आणि ट्रिगरिंगचे नियंत्रण ठेवते. या रजिस्टरमध्ये खालील महत्त्वाचे बिट्स असतात:

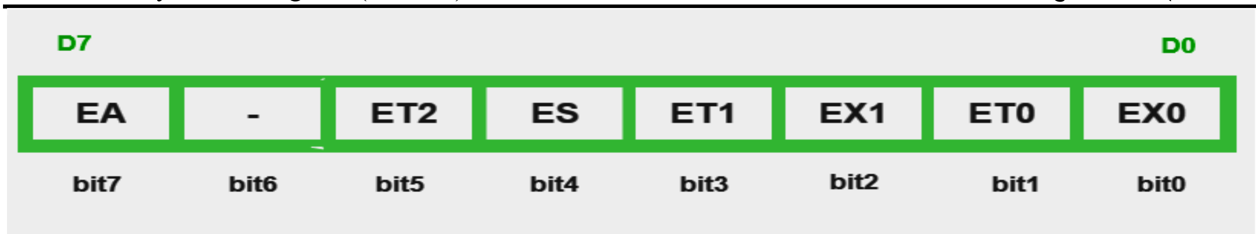


Figure.2.5.1 IE (Interrupt Enable) Register

EA (Global Interrupt Enable/Disable) – हे सर्व इंटरप्टसाठी सक्षम (Enable) किंवा अक्षम (Disable) करण्याचे नियंत्रण ठेवते.

0 – सर्व इंटरप्ट विनंत्या अक्षम (Disable) केल्या जातात.

1 – सर्व इंटरप्ट विनंत्या सक्षम (Enable) केल्या जातात.

ES (Serial Communication Interrupt Enable) – हे बिट सिरियल कम्युनिकेशन (UART) साठी इंटरप्ट सक्षम किंवा अक्षम करते.

1 – UART प्रणालीद्वारे इंटरप्ट सक्षम.

0 – UART प्रणालीद्वारे इंटरप्ट जनरेट होऊ शकत नाही.

ET0 (Timer 0 Interrupt Enable) – हे बिट Timer 0 साठी इंटरप्ट सक्षम किंवा अक्षम करते.

1 – Timer 0 इंटरप्ट सक्षम.

0 – Timer 0 इंटरप्ट जनरेट होऊ शकत नाही.

ET1 (Timer 1 Interrupt Enable) – हे बिट Timer 1 साठी इंटरप्ट सक्षम किंवा अक्षम करते.

1 – Timer 1 इंटरप्ट सक्षम.

0 – Timer 1 इंटरप्ट अक्षम.

EX0 आणि EX1 (External Interrupt 0 आणि External Interrupt 1 Enable) – ही बिट्स बाह्य उपकरणांमधून (External Devices) इंटरप्ट सक्षम किंवा अक्षम करतात.

EX0 – External Interrupt 0 सक्षम किंवा अक्षम करते.

0 – INT1 पिनवरील लॉजिक स्थिती बदलल्याने इंटरप्ट जनरेट होणार नाही.

1 – INT1 पिनवरील स्थिती बदलल्यावर बाह्य इंटरप्ट सक्षम.

EX1 – External Interrupt 1 सक्षम किंवा अक्षम करते.

0 – INT0 पिनवरील लॉजिक स्थिती बदलल्याने इंटरप्ट जनरेट होणार नाही.

1 – INT0 पिनवरील स्थिती बदलल्यावर बाह्य इंटरप्ट सक्षम.

IT0 आणि IT1 (बाह्य इंटरप्ट प्रकार - External Interrupt Type)

IT0 आणि IT1 ही बिट्स बाह्य इंटरप्ट ट्रिगर प्रकार (level-triggered किंवा edge-triggered) निश्चित करतात.

IP (Interrupt Priority) रजिस्टर: इंटरप्ट केव्हा प्राप्त होईल हे निश्चित करता येत नाही. जर एकापेक्षा जास्त इंटरप्ट सक्षम (Enable) असतील, तर एका इंटरप्ट प्रक्रियेदरम्यान दुसऱ्या इंटरप्टची विनंती येऊ शकते. अशा परिस्थितीत, मायक्रोकंट्रोलरला कोणत्या इंटरप्टला प्रथम प्रतिसाद द्यायचा यासाठी एक प्राधान्यक्रम (Priority List) दिला जातो.

मायक्रोकंट्रोलर रीसेट झाल्यावर सर्व प्रक्रिया थांबतात आणि तो पुन्हा सुरुवातीपासून कार्यरत होतो. फक्त Reset इंटरप्ट प्राधान्य 1 (Priority 1) अक्षम करू शकतो. Reset आणि Priority 1 दोघेही Priority 0 अक्षम करू शकतात.

IP (Interrupt Priority) रजिस्टर हे सध्याच्या कार्यरत इंटरप्ट्सपैकी कोणते अधिक महत्त्वाचे आहे हे दर्शवते. कार्यक्रमाच्या सुरुवातीसच इंटरप्टचे प्राधान्य निश्चित केले जाते. जर उच्च प्राधान्य असलेल्या (Higher Priority) इंटरप्टची विनंती आली आणि त्याच वेळी दुसरा इंटरप्ट चालू असेल, तर सध्याच्या इंटरप्टला थांबवून उच्च प्राधान्य असलेला इंटरप्ट प्रथम हाताळला जातो. जर दोन डिफरेंट प्रायोरिटी असलेल्या इंटरप्ट रिक्वेस्ट्स (requests) एकाच वेळी आल्या, तर उच्च प्राधान्य असलेला इंटरप्ट प्रथम हाताळलाजातो.परंतु जर दोन समान प्राधान्य असलेल्या (Same Priority Level) इंटरप्ट रिक्वेस्ट्स (requests) सलग आल्या, तर दुसऱ्या विनंतीला पहिल्या प्रक्रिया पूर्ण होईपर्यंत थांबावे लागते.

---	---	PT2	PS	PT1	PX1	PT0	PX0

Figure. 2.5.2 IP (Interrupt_Priority) Register

Bit0 (PX0) - External 0 इंटरप्ट प्राधान्य बिट

0 - बाह्य 0 इंटरप्टला कमी प्राधान्य देते.

1 - बाह्य 0 इंटरप्टला उच्च प्राधान्य देते.

Bit1 (PT0) - Timer 0 इंटरप्ट प्राधान्य बिट

0 - Timer 0 इंटरप्टला कमी प्राधान्य देते.

1 - Timer 0 इंटरप्टला उच्च प्राधान्य देते.

Bit2 (PX1) - External 1 इंटरप्ट प्राधान्य बिट

0 - बाह्य 1 इंटरप्टला कमी प्राधान्य देते.

1 - बाह्य 1 इंटरप्टला उच्च प्राधान्य देते.

Bit3 (PT1) - Timer 1 इंटरप्ट प्राधान्य बिट

0 - Timer 1 इंटरप्टला कमी प्राधान्य देते.

1 - Timer 1 इंटरप्टला उच्च प्राधान्य देते.

Bit4 (PS) - Serial इंटरप्ट प्राधान्य बिट

0 - Serial इंटरप्टला कमी प्राधान्य देते.

1 - Serial इंटरप्टला उच्च प्राधान्य देते.

Bit5, Bit6, Bit7 - Reserved Bits-हे बिट्स राखीव (Reserved) आहेत आणि भविष्यातील वापरासाठी राखून ठेवलेले असतात.

EXERCISE:

TLO 2.1: List the various data types in Embedded 'C'.

1. List data types in Embedded C with their value range. (R)
2. List logical operators in C with one example of each. (R)
3. List arithmetic operations in C with example of each(R)
4. Write C language program to shift bits with left shift (<<) and right shift **(>>)** (A)
5. Write C language program to swap two numbers without using a temporary variable, using the XOR operator. (A)
6. Write a C language program to toggle a specific bit in an 8-bit number. (A)

TLO 2.2: Develop embedded 'C' program for arithmetic, logical, and data transfer operations.

1. Write an Embedded C program for multiplying two 8-bit numbers and send result on Port 1. (A)
2. Write Embedded C program to add two 16-bit numbers and store the result in a specific memory location. (A)
3. Explain logical shift operations in embedded programming and example using Embedded C.(U)
4. Write logical operators in C for AND, OR, EX-OR and NOT for 8051 and state one example each. (A)
5. Write C language program to read P1 and store the one's complement of P1 to P2. (A)
6. Write C language program to transfer 10 bytes from array A to array B.(A)
7. Write C language program to toggle only bit P2.1 with continuously without disturbing rest of the bits P2. (A)
8. Write C language program to send values 00-FF to port 2. (A)

TLO 2.3: Develop embedded 'C' program using timer.

1. Write a program to conFIGUREure Timer 0 in Mode 1 for generating a delay of 1ms. Assume a crystal frequency of 11.0592 MHz. (A)
2. List different timer modes of 8051 microcontroller and describe mode 2 with neat sketch. (R)
3. Write C language program to generate a square wave of 5KHz on pin P1.0 by using timer 0 and mode 1. Assume crystal frequency is 12MHz. (A)
4. Explain the modes of timer in 8051. (U)
5. Draw the format of TMOD register of 8051 and describe the function of each bit. (U)
6. Draw the format of timer control register TCON of 8051. (A)

TLO 2.4: Develop embedded 'C' program for serial transmission and reception.

1. Write an Embedded C program to transmit the message "HELLO" serially at a baud rate of 9600, 8 data bits and 1 stop bit. (A)
2. Draw the format of SCON register and explain function of each. (A)
3. Write a program to receive data serially and display it on Port 1 using Embedded C.(A)
4. Write program in Embedded C language to display "WELCOME"(A)

TLO 2.5: Explain the 8051 interrupts.

1. List various interrupts in 8051 microcontroller along with their priorities and vector locations. (U)
2. Draw IE register format and explain function of each bit in 8051 microcontroller. (U)
3. Draw IP register format and explain function of each bit in 8051 microcontroller.(U)
4. Differentiate between software interrupt and hardware interrupt.(U)
5. Write an Embedded C program to generate an interrupt when a switch is pressed.(A)

युनिट 3. कम्युनिकेशन स्टँडर्ड आणि प्रोटोकॉल्स
(Communication Standards and Protocols)

Course Outcomes: Interpret the communication protocols of embedded systems.

सिद्धांत शिक्षण निष्पत्ती (Theory Learning Outcomes)

TLO 3.1: सिरीयल व पॅरॅलल कम्युनिकेशन यामधील तसेच सिंक्रोनस (Synchronous) व असिंक्रोनस (Asynchronous) कम्युनिकेशन यामधील तुलना करा.
TLO 3.2: सिरीयल कम्युनिकेशन प्रोटोकॉल स्पष्ट करा.
TLO 3.3: प्रगत सिरीयल प्रोटोकॉल्सची महत्त्वपूर्ण वैशिष्ट्ये वर्णन करा.

3.1 डेटा कम्यूनिकेश्न् प्रकार:

एम्बेडेड सिस्टीममध्ये (embedded system), विविध उप-सिस्टीम (sub system) आणि बाह्य उपकरणांशी (external devices) संवाद साधण्यासाठी कम्यूनिकेश्न्ची नेहमीच आवश्यकता असते. एम्बेडेड सिस्टीममध्ये आवश्यक असलेली कम्यूनिकेश्न् इंटरफेस(interface) सिरीयल(serial), पॅरलल(parallel) किंवा वायरलेस(wireless) असू शकते, हे उपकरणावर अवलंबून असते. जर एम्बेडेड सिस्टीममधील उपकरणे सिरीयल असतील, तर सिरीयल कम्यूनिकेश्न् इंटरफेस जसे की RS-232, I²C, CAN, USB, SPL, SSP इत्यादी वापरले जाऊ शकतात. जर उपकरणे पॅरलल असतील, तर पॅरलल कम्यूनिकेश्न् इंटरफेस जसे की ISA, PCI, PCI-X इत्यादी वापरले जाऊ शकतात. पण काही वेळा वायरलेस इंटरफेस देखील आवश्यक असू शकतात, अशा परिस्थितीत वायरलेस प्रोटोकॉल्स(protocols) जसे की IrDA, Bluetooth, Zigbee, IEEE802.11 इत्यादी वापरले जाऊ शकतात. ट्रान्समिशन मोड (Transmission Mode) म्हणजे दोन उपकरणांदरम्यान डेटा ट्रान्सफर करण्याची पद्धत. याला कम्यूनिकेश्न् मोड (Communication Mode) देखील म्हटले जाते. बस(Bus)आणि नेटवर्क्स(Networks)अशा प्रकारे डिझाइन केले जातात की, इंटरकनेक्टेड उपकरणांमध्ये संवाद होऊ शकतो. डेटा सोर्स (Source) पासून डेस्टिनेशन (Destination) कडे विविध पद्धतींनी ट्रान्समिट केला जाऊ शकतो. डेटा ट्रान्समिशनचे विविध प्रकार खालीलप्रमाणे आहेत:

1. सिरीयल आणि पॅरलल कम्यूनिकेश्न (Serial and Parallel Communication).
2. असिंक्रोनस आणि सिंक्रोनस कम्यूनिकेश्न (Asynchronous and Synchronous Communication)

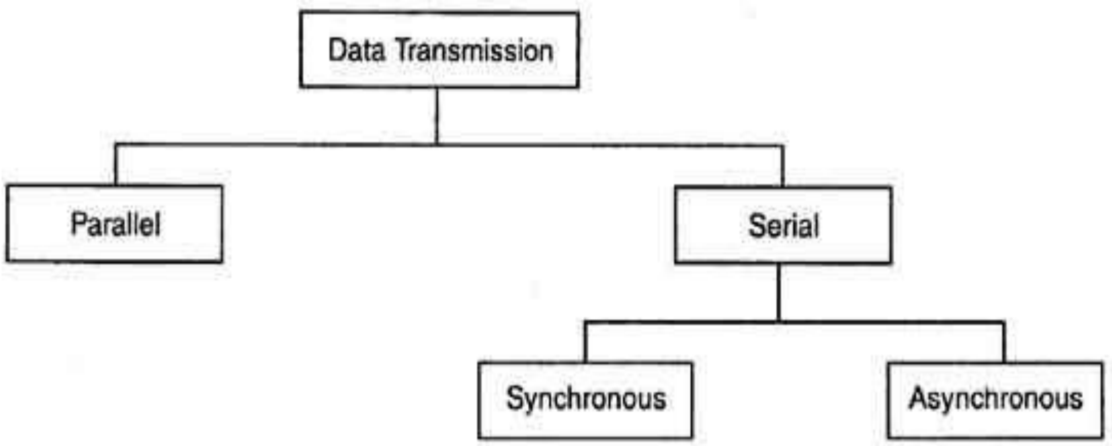


Figure.3.1.1 डेटा कम्यूनिकेश्न् प्रकार

3.1.1 सिरीयल आणि पॅरलल कम्यूनिकेश्न्: (Serial and Parallel Communication)

1) सिरीयल कम्यूनिकेश्न्:

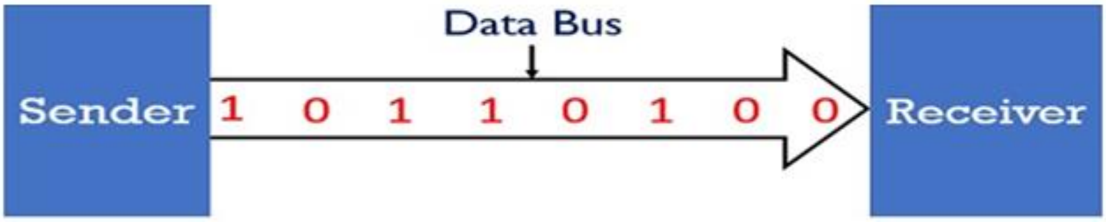


Figure.3.1.2 सिरीयल कम्यूनिकेश्न्

सिरीयल कम्यूनिकेश्न्ही मायक्रोकंट्रोलर (microcontroller) किंवा कम्प्युटर आणि पेरिफेरल डिव्हाइस(peripheral device) दरम्यान डेटा ट्रान्सफर करण्याची सर्वात सामान्य पद्धत आहे. सिरीयल कम्यूनिकेश्न्मध्ये, सेंडर (sender) एकावेळी एक बिट ट्रान्समिट करतो आणि तो एकच कम्यूनिकेश्न् लाईनवर रिसीव्हरकडे(receiver) पाठवला जातो, जसे की FIGURE3.1.2 मध्ये दर्शवले आहे. जेव्हा डेटा ट्रान्सफर रेट खूप कमी असतो आणि डेटा लांब अंतरावर ट्रान्सफर केला जातो, ज्यामध्ये केबलची किंमत आणि सिंक्रोनायझेशन (synchronization) आवश्यक असते, तेव्हा सिरीयल कम्यूनिकेश्न् वापरले जाते. सिरीयल कम्यूनिकेश्न् प्रसिद्ध आहे कारण पेरिफेरल डिव्हाइसला कम्प्युटरशी जोडण्यासाठी किंवा दोन कम्प्युटरना एकत्र जोडण्यासाठी केबलपेक्षा इतर हार्डवेअर आवश्यक नसते. सिरीयल कम्यूनिकेश्न् सिंक्रोनस किंवा असिंक्रोनस असू शकते, हे पेरिफेरलच्या स्पीडवर अवलंबून असते.

2) पॅरलल कम्यूनिकेश्न् :

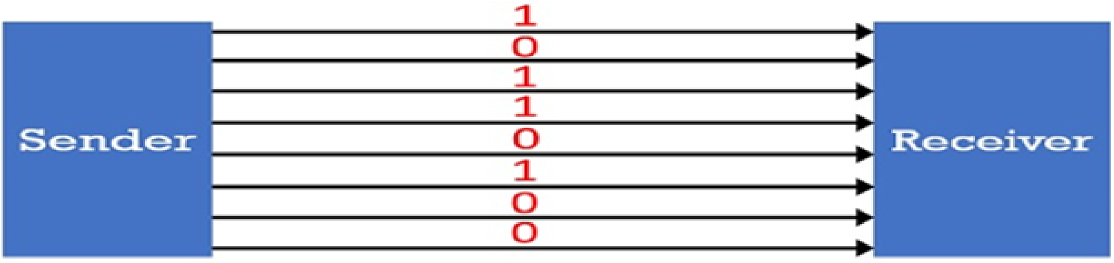


Figure.3.1.3 पॅरलल कम्यूनिकेश्न्

पॅरलल डेटा ट्रान्सफर मध्ये, डेटा च्या सर्व बिट्स एका वेळी एका सिंगल क्लॉक पल्समध्ये (clock pulse) ट्रान्समिट केले जातात, जसे की FIGURE 3.1.3 मध्ये दर्शवले आहे. पॅरलल डेटा कम्युनिकेशन मध्ये, प्रत्येक बिटसाठी वेगळ्या लाईन आवश्यक असते, उदाहरणार्थ 8-बिट डेटा ट्रान्सफर करण्यासाठी 8 वेगळ्या रेषा लागतात आणि डेटा बाइटसह पॅरिटी(parity) बिट ट्रान्सफर करण्यासाठी एक वेगळ्या रेषा, म्हणजे 9 रेषा लागतात, जी एरर (error) डिटेक्शन बिट म्हणून काम करते. डेटा ट्रान्सफर रेट (rate)सिरीयल डेटा ट्रान्सफरच्या तुलनेत खूप जलद असतो. पॅरलल डेटा ट्रान्सफर फक्त छोट्या अंतरासाठी वापरता येतो, कारण रिसीव्हर(receiver) आणि ट्रान्समीटर(transmitter) यांच्यातील अंतर कमी असले पाहिजे. उदाहरणार्थ, प्रिंटर हे पॅरलल डेटा ट्रान्सफर वापरणारे एक डिव्हाइस आहे.

Table 3.1 सिरीयल आणि पॅरलल कम्यूनिकेश्न् मधील तुलना(Comparison)**:**

तुलनेसाठी आधार (Basis for Comparison)	**सिरीयल कम्यूनिकेश्न्**	**पॅरलल कम्यूनिकेश्न्**
डेटा ट्रांसमिशन स्पीड	स्लो	फास्ट
कम्यूनिकेश्न् लिंकची संख्या	एक	अनेक (Multiple)
ट्रॅन्स्मिट् बिट/क्लॉकची संख्या	फक्त एक बिट	n नंबर लिंक n बिट कॅरी करतात
किंमत (Cost)	कमी	जास्त
क्रॉसस्टॉक (Crosstalk)	अस्तित्वात नसते (Not present)	अस्तित्वात असते (Present)
सिस्टम अपग्रेडेशन (System Up-gradation)	सोपे	अवघड
ट्रान्समिशन प्रकार	फुल-डुप्लेक्स (Full duplex)	हाफ डुप्लेक्स (Half duplex)
सोयीस्कर (SuiTABLE for)	जास्त अंतर (Long distance)	लहान अंतर (Short distance)
हाय फ्रिक्वेन्सी ऑपरेशन (High frequency operation)	अधिक प्रभावी (More efficient)	कमी प्रभावी (Less efficient)

3.1.2 असिंक्रोनस आणि सिंक्रोनस कम्यूनिकेश्न् :

1.असिंक्रोनस कम्यूनिकेश्न्

असिंक्रोनस सिरीयल कम्यूनिकेश्न् मध्ये, प्रत्येक बाइट किंवा कॅरेक्टर(character) पाठवण्यापूर्वी एक स्टार्ट बिट पाठवला जातो आणि प्रत्येक डेटा बाइटनंतर एक स्टॉप बिट पाठवला जातो, जसे की FIGURE3.1.4 मध्ये दर्शवले आहे. स्टार्ट बिट ट्रान्समीटर आणि रिसीव्हर सिंक्रोनाइझ करतो. स्टॉप बिट बाइट किंवा कॅरेक्टरचा(character) शेवट दर्शवतो आणि पुढील कॅरेक्टरच्या रिसेप्शनसाठी रिसीव्हिंग मॅकेनिझम सेट करण्यासाठी देखील वापरला जातो. FIGURE3.1.4 मध्ये, सुरुवातीला स्टार्ट बिट पाठवले जाते, त्यानंतर आठ डेटा बिट्स, पॅरिटी बिट आणि एक स्टॉप बिट पाठवला जातो, ज्यामुळे 10-बिट कॅरेक्टर फ्रेम तयार होते. स्टॉप बिट नंतर, लाईन अनिश्चित कालावधीसाठी रिकामी राहू शकते, किंवा दुसरे कॅरेक्टर लगेच ट्रान्समिट केले जाऊ शकते. सिरीयल कम्युनिकेशनची महत्त्वाची वैशिष्ट्ये म्हणजे बॉड रेट, डेटा बिट्स, स्टॉप बिट्स, आणि पॅरिटी.

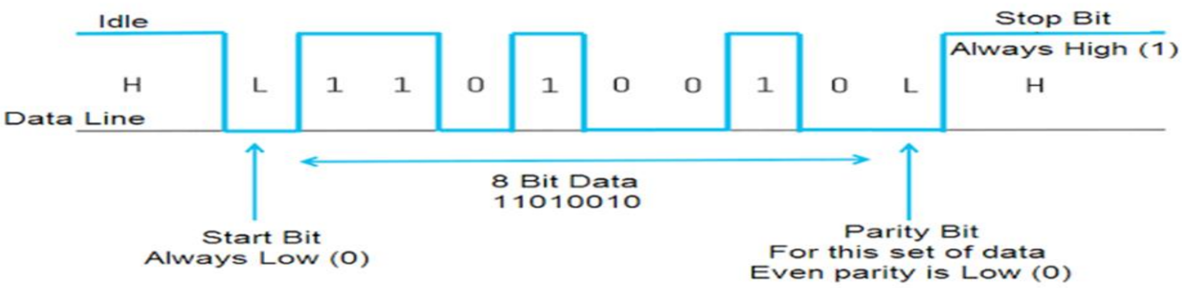


Figure.3.1.4 फ्रेम फॉरमॅट असिंक्रोनस डेटा

दोन उपकरणांमधील कम्युनिकेशनदरम्यान, खालील पॅरामीटर्स जुळवणे आवश्यक आहे

(a) **बॉड रेट (Baud Rate)**:
डेटा कम्युनिकेशनची गती, जी बिट्स प्रतिसेकंद मध्ये मोजली जाते. उदाहरणार्थ, 300 बॉड म्हणजे 300 बिट्स प्रति सेकंद.

(b) **डेटा बिट्स (Data Bits)**:
डेटा पॅकेटमध्ये ट्रान्समिशन दरम्यान प्रत्यक्ष डेटा बिट्सची संख्या डेटा पॅकेटमध्ये व्हॅल्यू 5, 7, किंवा 8 बिट्स असतात.

(c) **स्टॉप बिट्स (Stop Bits)**:
एक पॅकेटच्या कम्युनिकेशनचा शेवट दर्शवणारा बिट. सामान्य स्टॉप बिट्सचे व्हॅल्यू 1, 1.5 आणि 2 बिट्स असतात

(d) **पॅरिटी (Parity)**:
सिरीयल कम्युनिकेशनमध्ये वापरली जाणारी साधी एरर चेकिंग पद्धत. पॅरिटीचे प्रकार इव्हन (Even) किंवा ऑड (Odd) असू शकतात, म्हणजेच एक इव्हन(Even) किंवा ऑड (Odd) लॉजिक हाय(logic high) बिट्स 1 असू शकतात.

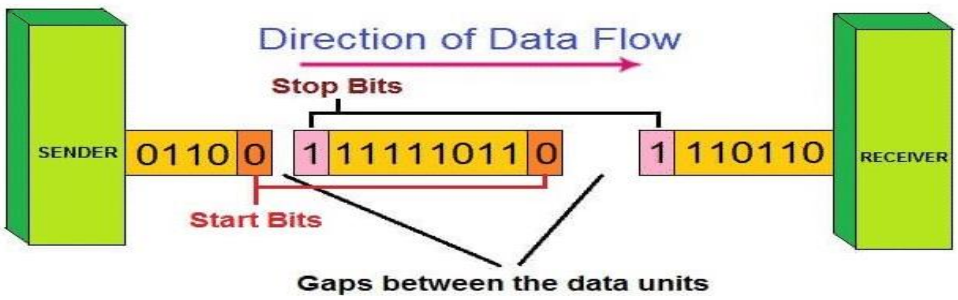


Figure.3.1.5 असिंक्रोनस कम्यूनिकेश्न्

स्टार्ट आणि स्टॉप बिट्स जोडल्यामुळे डेटा बिट्सची संख्या वाढते. त्यामुळे असिंक्रोनस ट्रान्समिशनमध्ये अधिक बँडविड्थ(Bandwidth) वापरला जातो. वेगवेगळ्या डेटा बाइट्सच्या ट्रान्समिशनदरम्यान एक रिकामी वेळ (Idle Time) असते. या रिकामी वेळेला गॅप (Gap) म्हणून ओळखले जाते. गॅप किंवा रिकामी वेळ वेगवेगळ्या अंतराने असू शकते. हा मेकॅनिझम असिंक्रोनस म्हणून ओळखला जातो, कारण बाइट स्तरावर सेंडर (sender) आणि रिसीव्हरचे(receiver) सिंक्रोनायझेशन आवश्यक नाही. पण प्रत्येक बाइटमध्ये, रिसीव्हरला आलेल्या बिट्सच्या इनकमिंग बिट स्ट्रीमसह सिंक्रोनाइझ असणे आवश्यक आहे

२) सिंक्रोनस कम्युनिकेशन:
सिंक्रोनस सीरियल कम्युनिकेशनमध्ये, दोन उपकरणे सुरुवातीला एकमेकांशी सिंक्रोनाइझ करतात आणि त्यानंतर सतत सिंक्रोनस पद्धतीने कॅरेक्टर्स पाठवतात.
सिंक्रोनस कम्युनिकेशनचा डेटा ट्रान्सफर दर असिंक्रोनस पद्धतीपेक्षा वेगवान असतो, कारण डेटा बाइटसाठी अतिरिक्त माहितीची आवश्यकता नाही.
सिंक्रोनस कम्युनिकेशन मध्ये, डेटा बाइट्स एकामागोमाग एक ट्रान्सफर केले जातात. सेंडर (sender) आणि रिसीव्हरला(receiver) सिंक्रोनाइझ करण्यासाठी, डेटा बाइट्स पाठवण्यापूर्वी पाठवणारा एक विशेष कॅरेक्टर पाठवतो, ज्याला "सिंक्रोनस कॅरेक्टर" असे म्हणतात. सुरुवातीला, ट्रान्समीटरची आउटपुट लाईन हाय असते, म्हणजेच ती मार्किंग स्थितीत असते.

सिंक्रोनस कॅरेक्टर सर्वप्रथम ट्रान्समीटरद्वारे पाठवला जातो आणि त्यानंतर डेटा बिट्स पाठवले जातात, ज्यामुळे ट्रान्समिशन सुरू होते. सिंक्रोनस ट्रान्समिशनमध्ये स्टार्ट आणि स्टॉप बिट्सचा वापर केला जात नाही. या पद्धतीमध्ये बिट स्ट्रीम लांब फ्रेम्समध्ये एकत्र केले जाते, FIGURE.3.1.6 ज्या फ्रेम्समध्ये अनेक बाइट्स असू शकतात.डेटा स्ट्रीममधील विविध बाइट्स दरम्यान कोणतीही गॅप नसते. स्टार्ट आणि स्टॉप बिट्सच्या अनुपस्थितीत, प्रत्येक बिटच्या ट्रान्समिशनच्या 'टायमिंग'द्वारे सेंडर आणि रिसीव्हर यांच्यात बिट सिंक्रोनायझेशन स्थापित केले जाते विविध बाइट्स लिंकवर गॅप न करता ठेवले जातात, त्यामुळे मूळ माहितीची पुनर्रचना करण्यासाठी बिट स्ट्रीमचे बाइट्समध्ये विभाजित करणे ही रिसीव्हरची जबाबदारी आहे. डेटा एरर फ्री(error free) प्राप्त करण्यासाठी, रिसीव्हर आणि सेंडर एकाच क्लॉक फ्रिक्वेन्सीवर कार्य करतात.

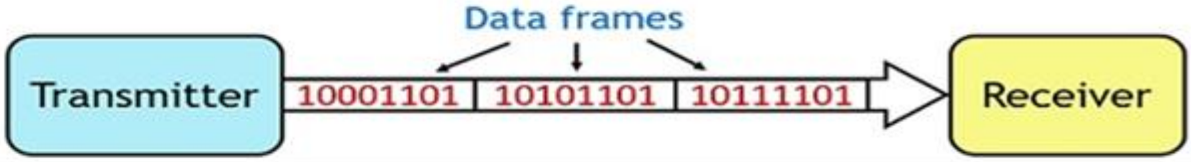


Figure.3.1.6 सिंक्रोनस कम्युनिकेशन

तुलनेसाठी आधार (Basis for Comparison)	**सिंक्रोनस कम्युनिकेशन**	**असिंक्रोनस कम्युनिकेशन**
क्लॉक पल्स (Clock pulse)	ट्रान्समीटर आणि रिसीव्हरमध्ये क्लॉकचा एक समान क्लॉक पल्स शेअर करतात.	ट्रान्समीटर आणि रिसीव्हरमध्ये एक समान क्लॉक पल्स शेअर करत नाहीत
ट्रांसमिशन स्पीड (Speed of	Fast	Comparatively slow.

Table 3.2 सिंक्रोनस आणि असिंक्रोनस कम्युनिकेशन मधील तुलना(comparision)

transmission)		
डेटा ट्रान्समिशनचे स्वरूप (Form of data transmission)	डेटा फ्रेम किंवा ब्लॉक्सच्या स्वरूपात पाठवला जातो.	डेटा बाइट किंवा कॅरेक्टरच्या स्वरूपात पाठवला जातो.
टाइम इन्टर्व्हल (Time interval)	स्थिर (Constant)	अनेकदा बदलणारे (Variable)
किंमत (Cost)	महाग (Expensive)	स्वस्त (less expensive)
इफ़िश्न्सि (Efficiency)	अधिक प्रभावी (More efficient)	कमी प्रभावी (Less efficient)
एक्सटर्नल क्लॉक गरज (Need of external clock)	गरज असते (Exist)	गरज नसते (Do not exist)

3.2 सीरियल कम्युनिकेशन स्टैंडर्ड(standard) RS232 (DB9):

RS-232 म्हणजेच रेकमेंडेड स्टँडर्ड 232 हे एका सिरियल पोर्टचे एक प्रकार आहे, जे बहुतेक कम्प्युटरमध्ये वापरले जाते. RS-232 इंटरफेस हा 25-पिन "D" कनेक्टर असतो, परंतु फक्त 22 पिन्सच वापरले जातात कारण ह्या पिन्सपैकी अनेक पिन्स सामान्य PC कम्युनिकेशन्ससाठी आवश्यक नसतात. म्हणूनच, बहुतेक नवीन PC मध्ये 9 पिन्स असलेला मेल् (male) D-टाइप कनेक्टर असतो. RS-232 हा सिंक्रोनस आणि अॅसिंक्रोनस दोन्ही मोड्सला सपोर्ट करतो, पण RS-232 डेटा ट्रान्समिशन सामान्यतः अॅसिंक्रोनस असतो. RS-232 चे वर्गीकरण डेटा टर्मिनल इक्विपमेंट (DTE) किंवा डेटा कम्युनिकेशन इक्विपमेंट (DCE) असे केले जाते, ज्याने कोणत्या वायरवर कोणता सिग्नल पाठविला जातो आणि प्राप्त केला जातो हे निश्चित होते. RS-232 इंटरफेस 15 मीटर पर्यंतच्या अंतरावर कार्य करू शकतो. RS-232 इंटरफेसचा डेटा रेट देखील मर्यादित आहे, आणि त्याचा जास्तीत जास्त डेटा रेट 19.2 k बोड (बिट्स प्रति सेकंद) आहे, काही वेळा अधिक कमी रेटवर डेटा ट्रान्समिशन केला जातो. RS-232 साठी डेटा सिग्नल्स हे -3 V आणि -25 V च्या दरम्यान असतात, जे लॉजिक 1 दर्शवतात. लॉजिक 0 हा +3 V आणि +25 V दरम्यानच्या व्होल्टेजने दर्शविला जातो. RS-232 इंटरफेसमधील कंट्रोल सिग्नल्स हे "ON" स्थितीत असतात जर त्यांचा व्होल्टेज +3 V आणि +25 V च्या दरम्यान असेल, आणि "OFF" स्थितीत असतात जर ते नकारात्मक, म्हणजेच -3 V आणि -25 V च्या दरम्यान असतील. डेटा बिट्स RS-232 वर सिरिअली पाठवले जातात, म्हणजेच प्रत्येक बिट एकामागून एक पाठवला जातो कारण प्रत्येक दिशा मध्ये फक्त एक डेटा लाइन असतो, म्हणून हे फुल डुप्लेक्स कम्युनिकेशनला सपोर्ट करते. डेटा बिट नंतर एक पॅरिटी बिट पाठवला जातो, जो सम किंवा विषम पॅरिटी असू शकतो, आणि शेवटी एक स्टॉप बिट पाठवला जातो जो एक किंवा दोन बिट्स लांबीचा असतो आणि तो विशिष्ट बाइटच्या शेवट दर्शवण्यासाठी वापरला जातो. RS-232 वर पाठवलेला डेटा साधारणपणे ASCII (अमेरिकन स्टँडर्ड कोड फॉर इन्फॉर्मेशन इंटरचेंज) वापरून पाठवला जातो. RS-232 स्पेसिफिकेशनमध्ये चार प्रकारच्या लाईन्स स्पष्ट केलेल्या आहेत, म्हणजेच डेटा, कंट्रोल, टाइमिंग आणि ग्राउंड.

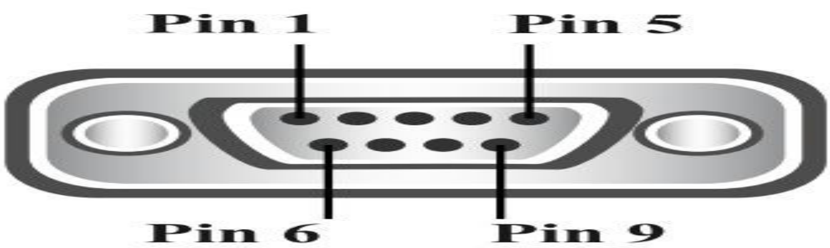


Figure.3.2.1 RS-232 कनेक्टर DB-9

RS-232 (DB9) पिन फंक्शन्स :

पिन 1: डेटा कॅरियर डिटेक्ट (DCD): हा कंट्रोल सिग्नल मोडेमद्वारे तयार केला जातो, ज्यामुळे कम्प्युटरला माहिती दिले जाते की त्याने दुसऱ्या मोडेमशी भौतिक(physical) कनेक्शन स्थापित (established) केले आहे. जर भौतिक कनेक्शन तुटले, तर ही सिग्नल लाइन स्थिती बदलते.

पिन 2: रिसीव्ह डेटा (RD किंवा RxD): ही लाइन डेटा प्राप्त करण्यासाठी वापरली जाते, जी मोडेमद्वारे कम्प्युटरला ट्रान्समिट केली जाते.

पिन 3: ट्रान्समिट डेटा (TD किंवा TxD): ही लाइन कम्प्युटरपासून मोडेमला डेटा ट्रान्समिट करण्यासाठी वापरली जाते.

पिन 4: डेटा टर्मिनल रेडी (DTR):हा सिग्नल कम्प्युटरद्वारे तयार केला जातो ज्यामुळे मोडेमला माहिती दिले जाते की कम्प्युटर डेटा घेण्यासाठी तयार आहे.

पिन 5: सिग्नल ग्राउंड: हा सिग्नल ग्राउंड पिन आहे.

पिन 6: डेटा सेट रेडी (DSR): हा सिग्नल मोडेमद्वारे तयार केला जातो, जो कम्प्युटरच्या DTR सिग्नलच्या रिस्पॉन्स(response) म्हणून असतो, आणि कम्प्युटरला हा सिग्नल लाइन मॉनिटर करण्याची आवश्यकता असते, जेणेकरून DTR सेट केल्यावर मोडेम चालू आहे का हे ओळखता येईल.

पिन 7: रिक्वेस्ट टू सेंड (RTS): हा सिग्नल कम्प्युटरद्वारे (DTE) तयार केला जातो, ज्यामुळे मोडेम DCE ला डेटा ट्रान्समिट करण्यासाठी माहिती दिले जाते. जर मोडेमला हा सिग्नल योग्य वाटला, तर तो CTS सिग्नल तयार करतो, आणि त्यानंतर कम्प्युटर डेटा ट्रान्समिट करायला सुरुवात करतो.

पिन 8: क्लियर टू सेंड (CTS): हा सिग्नल मोडेमद्वारे RTS सिग्नल जनरेट केला जातो, ज्यामुळे कम्प्युटरला माहिती दिले जाते की आता तो डेटा ट्रान्समिट करू शकतो.

पिन 9: रिंग इंडिकेटर (RI): हा सिग्नल DCE (डेटा कम्युनिकेशन इक्विपमेंट) कडून DTE (डेटा टर्मिनल इक्विपमेंट) कडे जातो, ज्यामुळे माहिती दिले जाते की एक इनकमिंग कॉल (incoming call) आहे, म्हणजेच टेलिफोन रिंग करत आहे आणि हा सिग्नल फक्त स्विच केलेल्या सर्किट कनेक्शनवर वापरला जातो.

3.2.2 RS-232 चा पिन आउट:

DTE -डेटा टर्मिनल इक्विपमेंट (टर्मिनल किंवा कम्प्युटर जो टर्मिनल म्हणून कार्य करतो)
DCE- डेटा कम्युनिकेशन इक्विपमेंट(मोडेम)

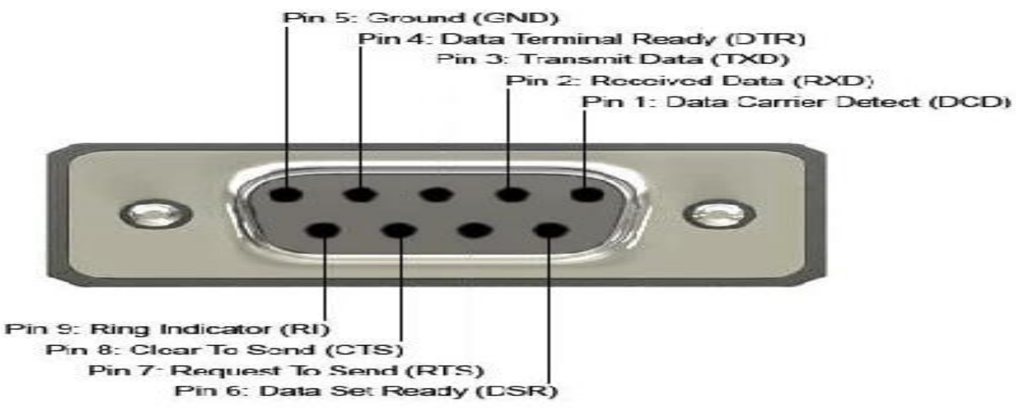


Figure.3.2.2 RS-232 पिन आउट डायग्राम(Diagram)

सिरिअल कम्युनिकेशन प्रोटोकॉल्स: एंबेडेड सिस्टिममध्ये सिरिअल कम्युनिकेशन इंटरफेस वापरले जातात, जसे की I²C, CAN, SPI इत्यादी

I²C (इंटर-इंटिग्रेटेड सर्किट):

परिचय (Introduction)**:** I²C म्हणजेच इंटर-इंटिग्रेटेड सर्किट हा एक दोन वायरचा इंटरफेस आहे, जो एक मल्टी-मास्टर सिरिअल सिंगल-एंडेड कंप्युटर बस आहे, जो लो-स्पीड पेरिफेरल्सना मदरबोर्ड, एंबेडेड सिस्टिम, किंवा सेल फोन किंवा इतर इलेक्ट्रॉनिक्ससह जोडण्यासाठी वापरला जातो, जसा की FIGURE3.3.1 मध्ये दर्शविला आहे.

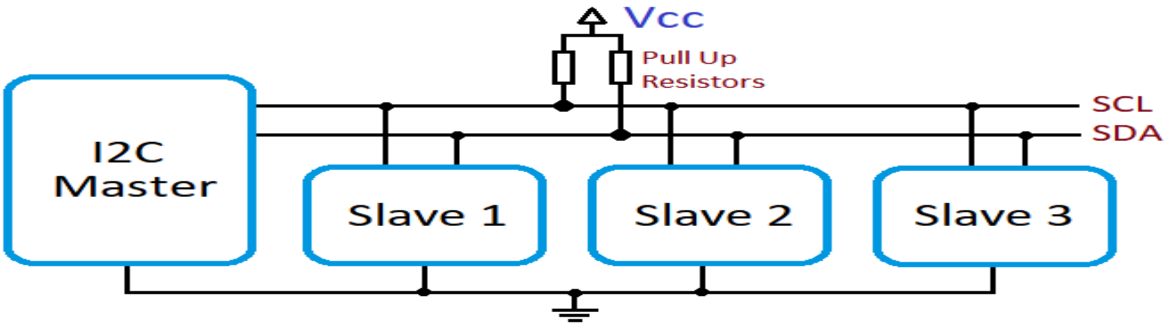


Figure.3.1. I²C मास्टर स्लेव्ह कॉन्फिगरेशन (I²C Master Slave Configurauration)

I²C मध्ये, मास्टर स्लेव्हकडे डेटा पाठवतो किंवा स्लेव्हकडून डेटा रिसीव्ह(receive) करतो, पण स्लेव्ह एकमेकांमध्ये डेटा ट्रान्सफर करत नाहीत. I²C मध्ये फक्त दोन बाय-डायरेक्शनल(bi-directional) लाइन, म्हणजे सिरिअल डेटा (SDA) आणि सिरिअल क्लॉक (SCL) लागतात जे डेटा एका डिव्हाइसपासून दुसऱ्या डिव्हाइसवर पाठवतात. प्रत्येक I²C डिव्हाइसला 7-बिट ॲड्रेसद्वारे अॅड्रेस केला जातो, ज्यामध्ये 16 रिझर्व्ह (reserved) केलेले ॲड्रेस असतात, त्यामुळे एकाच बसवर जास्तीत जास्त 112 नोड्स आपसात कम्यूनिकेट् करू शकतात. I²C डिव्हाइस जे कम्यूनिकेश्न्ची सुरुवात करते त्याला मास्टर म्हणतात, जो क्लॉक सिग्नलचे नियंत्रण ठेवतो, आणि मास्टर एकाच वेळी 127 स्लेव्हसना अॅड्रेस करू शकतो. जो डिव्हाइस मास्टरद्वारे अॅड्रेस केला जातो त्याला स्लेव्ह म्हटले जाते. कॉमन (common) I²C बस स्पीड ह्या प्रमाणे असतात: स्टँडर्ड मोडमध्ये 100 Kbit/s, लो-स्पीड मोडमध्ये 10 Kbit/s, फास्ट मोडमध्ये 400 Kbit/s, आणि हाय स्पीड मोडमध्ये 3.4 Mbps.

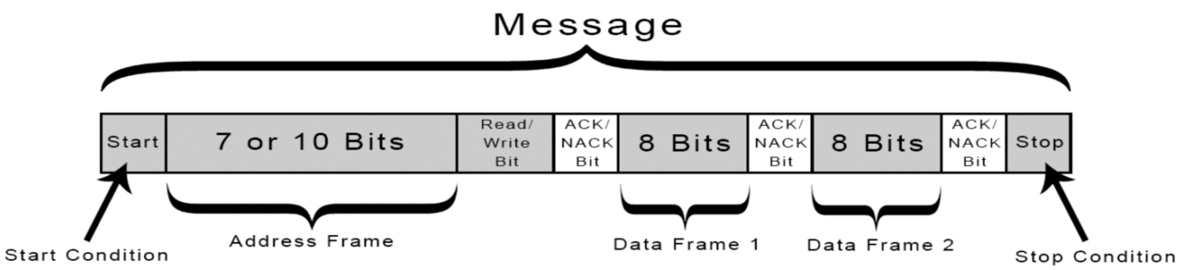


Figure.3.2 : I²C चे फ्रेम फॉरमॅट (Frame format)

Table 3.3.2: फ्रेम फॉरमॅट I²C चे बिट डिस्क्रिप्श्न् (description)

फील्ड आणि त्याची लांबी	डिस्क्रिप्शन
1st फील्ड चे 1 बिट	हे UART मध्ये स्टार्ट बिटसारखे आहे
2nd फील्ड चे 7 बिट	हे ॲड्रेस फील्ड आहे. ते स्लेव्ह ॲड्रेस डिफाईन करते, जो मास्टरद्वारे डेटा फ्रेममध्ये पाठवला जातो.

3rd फील्ड चे 1 कंट्रोल बिट	ते R/W सायकल चालू आहे की नाही हे डिफाईन करते.
4th फील्ड चे 1 कंट्रोल बिट	सध्याचा डेटा अक्नॉलिज्मन्ट् आहे की नाही हे डिफाईन करते.
5th फील्ड चे 8 बिट	हे IC डिव्हाइस डेटा बाइटसाठी आहे.
6th फील्ड चे 1- बिट	ते एक NACK आहे (कोणतीही अक्नॉलिज्मन्ट् नाही). जर ॲक्टीव असेल तर स्लेव्हकडून ACK ची आवश्यकता नाही, अन्यथा ACK ची आवश्यकता आहे..
7th फील्ड चे 1 बिट	UART मधील STOP बिट सारखेच.

I²C चे फ़ीचर्:

a. फक्त दोन बस लाइन लागतात, म्हणजे सिरिअल डेटा लाइन (SDA) आणि सिरिअल क्लॉक लाइन (SCL).
b. स्ट्रिक्ट् (Strict) बाऊड रेटची आवश्यकता नाही, कारण मास्टर बस क्लॉक तयार करतो, जो विविध डेटा स्पीड सपोर्ट करतो: स्टँडर्ड (100 kbps), फास्ट (400 kbps) आणि हाय स्पीड (3.4 Mbps) कम्युनिकेशन्स.
c. सर्व डिव्हाइसेससाठी साधे मास्टर/स्लेव्ह कॉन्फिगरेशन.
d. प्रत्येक डिव्हाइस जे बसला कनेक्ट केलेले आहे, ते सॉफ़्टवेअरद्वारे ॲड्रेस केले जाऊ शकते आणि प्रत्येकाला एक यूनीक्(unique) ॲड्रेस असतो.
e. PC एक खरा (true) मल्टी-मास्टर बस आहे, जो आर्बिट्रेशन (arbitration)आणि कोलिशन डिटेक्शन (collision detection) प्रव़ाइड्(providing) करतो.
f. इनबिल्ट कोलिशन डिटेक्शन.
g. 10-बिट ॲड्रेसिंग.
h. मल्टी-मास्टर सपोर्ट.
i. डेटा ब्रॉडकास्ट.

I²C सिरिअल बसचे ॲप्लिकेशन्स:

a. कॉम्प्युटिंग (Computing):
    a. सर्व्हर/स्टोरेज सर्व्हे,
    b. फॅन कंट्रोल,
    c. पॉवर सप्लाय,
    d. व्होल्टेज ट्रान्सलेशन.
b. कम्युनिकेशन (Communication):
    a. नेटवर्किंग लाइन कार्ड,
    b. पॅसिव ऑप्टिकल नेटवर्क,
    c. राऊटर.
c. औद्योगिक (इन्डस्ट्रिअल्) (Industrial):
    a. फॅक्टरी ऑटोमेशन,
    b. ऍक्सेस/अलार्म सिस्टिम्स,
    c. व्हिडीओ, LCD आणि LED डिस्प्ले साईन,
    d. हॉटेल/मोटेल मॅनेजमेंट सिस्टिम्स,
    e. आपत्कालीन लायटिंग/एक्झिट साईन मॉनिटर करणे,
    f. सामान्य उद्देशीय I/O (GPIO).
d. मोबाइल (Mobile):
    a. LCD किंवा कॅमेरा कंट्रोल,
    b. कीपॅड कंट्रोल,

c. गेमिंग/LED साईन.

CAN (कंट्रोलर एरिया नेटवर्क) प्रोटोकॉल:

परिचय (Introduction): कंट्रोलर एरिया नेटवर्क (CAN) हा एक सिरिअल नेटवर्क आहे जो मूळतः ऑटोमोटिव्ह इंडस्ट्रीसाठी डिझाइन केला गेला होता, पण आता तो औद्योगिक ऑटोमेशन तसेच इतर ऐप्लिकेश्न्साठी एक लोकप्रिय(popular) बस बनला आहे.
कंट्रोलर एरिया नेटवर्क (CAN-बस) हे वेहिकल (vehicle) बस आहेत, जे मायक्रोकंट्रोलर आणि डिव्हाइसेसना वेहिकलमध्ये एकमेकांशी कम्यूनिकेट् करण्याची परवानगी(permission) देतात, त्यासाठी होस्ट(host)कंप्युटरची आवश्यकता नाही.
हे दोन वायरचा, हाफ डुप्लेक्स, हाय-स्पीड नेटवर्क सिस्टम आहे, जसा की FIGURE3.3.3 मध्ये दाखवले आहे, आणि शॉर्ट मेसेजेस वापरून हाय-स्पीड ऐप्लिकेश्न्साठी योग्य आहे, म्हणून याला मेसेज-आधारित(based) प्रोटोकॉल म्हटले जाते.

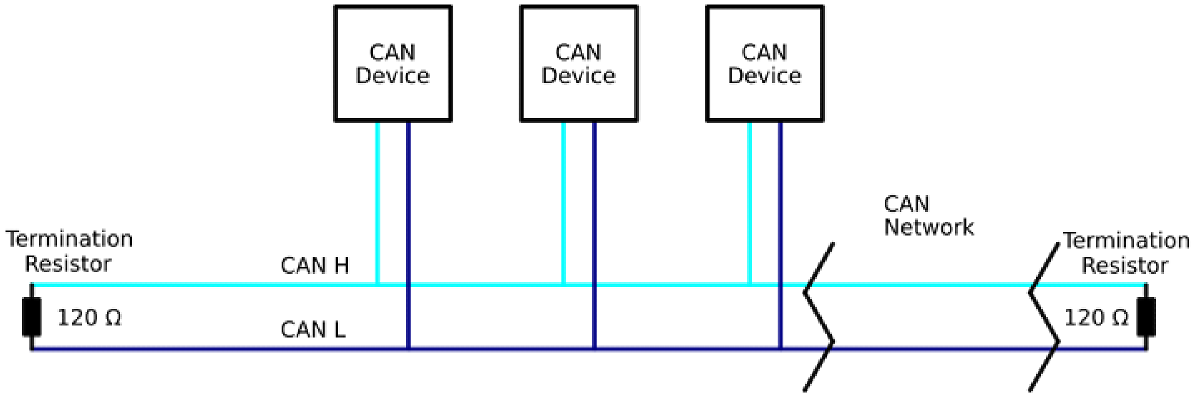


Figure.3.3.3 CAN-बस

हे हाय स्पीडच्या कम्युनिकेशनसाठी वापरले जाते, ज्याची स्पीड 1 Mbits/sec पर्यंत असू शकते, ज्यामुळे रिअल-टाइम कंट्रोल शक्य होतो. याशिवाय, एरर कनफाइनमेंट (confinement) आणि एरर डिटेक्शन (detection)हे त्याला नॉइज क्रिटिकल वातावरणात अधिक विश्वासार्ह (reliable) बनवतात.
CAN हे एक मल्टी-मास्टर नेटवर्क आहे जे इलेक्ट्रॉनिक कंट्रोल युनिट्स (ECUs) कनेक्ट करण्यासाठी वापरले जाते आणि यामध्ये CSMA/CD+AMP (कॅरियर सेंस मल्टिपल ॲक्सेस/कोलिशन डिटेक्शन विथ आर्बिट्रेशन ऑन मेसेज प्रायोरिटी) वापरले जाते. CAN नोड मेसेज पाठवण्यापूर्वी बस व्यस्त(busy) आहे का ते तपासते आणि कोलिशन डिटेक्शन (collision detection) देखील वापरते. प्रत्येक नोड मेसेज पाठवू शकतो आणि रिसीव्ह करू शकतो, परंतु एकाच वेळी नाही, कारण ते हाफ डुप्लेक्स कम्युनिकेशनला सपोर्ट करते.
एक मेसेज एका आयडी (आयडेंटिफायर) पासून बनलेला असतो, जो मेसेजची प्रायोरिटी दर्शवतो, आणि त्यामध्ये आठ डेटा बाइट्सपर्यंत असू शकतात.

CAN बसवरील कोणत्याही नोडमधून पाठवलेले डेटा मेसेज त्या नोडचा अड्रेस् किंवा कोणत्याही इन्टेन्ड् रिसीव् करणाऱ्या नोडचा अड्रेस् समाविष्ट करत नाहीत. मेसेजची कन्टेन्ट् (content) एक यूनीक् आयडेंटिफायर असतो जो नेटवर्कवरील सर्व नोड्ससाठी विशिष्ट असतो. हा मेसेज सिरिअली बसवर पाठवला जातो आणि नॉन-रिटर्न-टू-झीरो (NRZ) एन्कोडिंगमध्ये असतो, जो सर्व नोड्सद्वारे ओळखला जातो. नेटवर्कवरील सर्व नोड्स हा मेसेज रिसीव्ह करतात आणि प्रत्येक नोड आयडेंटिफायर तपासतो जेणेकरून ते तपासू शकेल की हा मेसेज, आणि त्याची महिती(content), त्या नोडसाठी संबंधित आहे का.

जर मेसेज संबंधित असेल, तर तो स्वीकारला(accepted)जाईल आणि प्रोसेस् केला जाईल, अन्यथा तो इग्नोर(ignored) केला जाईल. सामान्यतः, सेन्सर्स, अॅक्ट्युएटर्स, आणि इतर कंट्रोल डिव्हायसेस डरेक्ट्लि CAN बसवर कनेक्ट केले जात नाहीत, पण ते होस्ट प्रोसेसर आणि CAN कंट्रोलरद्वारे कनेक्ट केले जातात.

जर दोन किंवा अधिक डिव्हायसेस एकाच वेळी मेसेज पाठवण्यास सुरूवात केली, तर ज्या मेसेजचा आयडी हाय (highss) प्रायोरिटी असतो, तो इतर नोड्सना कमी (less)प्रायोरिटी असलेल्या आयडीसह ओव्हरराईड करतो, त्यामुळे फक्त डॉमिनंट मेसेज बसवर राहतो आणि सर्व नोड्सद्वारे रिसीव्ह केला जातो. म्हणून, याला प्रायोरिटी बस आर्बिट्रेशन म्हणतात, जिथे लेस् (less)आयडी असलेले मेसेज हाय (high)प्रायोरिटीचे मानले जातात आणि सर्वप्रथम ट्रान्समिट केले जातात. प्रत्येक नोडला होस्ट प्रोसेसर, CAN कंट्रोलर आणि ट्रान्ससीव्हरची आवश्यकता असते.

दोन स्पेसिफिकेशन्स वापरात आहेत:

1. **CAN 2.0A:**
   हे 11-बिट मेसेज आयडेंटिफायर असलेले बेसिक्(basic) किंवा स्टैन्डड्(standard) CAN म्हणून ओळखले जाते, आणि हे 250 Kbps च्या जास्तीत जास्त स्पीडवर ऑपरेट् करते. हे ISO 59.2 द्वारा स्टैन्डडाइज़् (standardized) केले आहे.
2. **CAN 2.0B:**
   हे 29-बिट मेसेज आयडेंटिफायर असलेले फ़ुल् किंवा इक्स्टेन्ड्(extended) CAN म्हणून ओळखले जाते, आणि हे 1 Mbps च्या जास्तीत जास्त स्पीडवर ऑपरेट् करते. हे ISO 11898 द्वारा स्टैन्डडाइज़् (standardized) केले आहे.
3. CAN विविध भौतिक मीडिआवर राबवता येऊ शकते कारण ड्रायव्हर्स ओपन कलेक्टर असतात आणि प्रत्येक नोड त्याच्याशी आणि इतर नोड्ससोबत कम्युनिकेशन करताना त्याला ऐकू शकते.
4. सर्वात कॉमन मीडिआ म्हणजे ट्विस्टेड पेअर 5V डिफरेंशियल सिग्नलिंग, जे हाय नॉईज असलेल्या वातावरणात ऑपरेट् करण्याची परवानगी देते.
5. CAN हाय स्पीडवर चालवताना बसच्या दोन्ही टोकांवर 120 ओहमच्या (ohm) रेजिस्टर्सने बस टमिनेट् करणे आवश्यक आहे, जेणेकरून रिफ़्लेक्शन् (reflection) टाळता येईल.

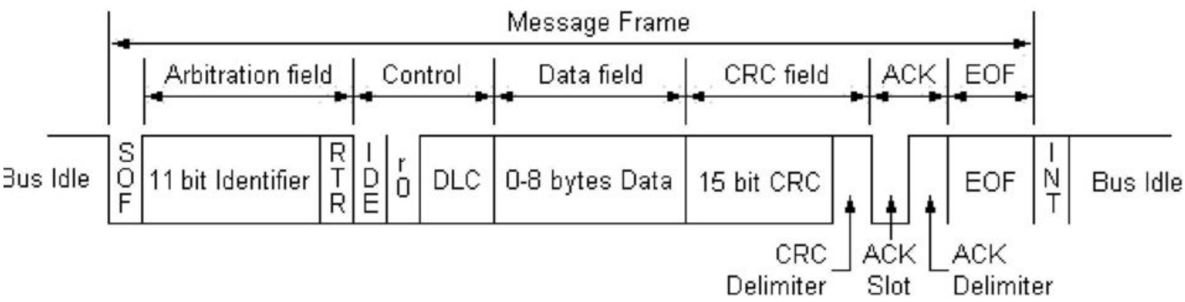


Figure.3.4: CAN 2.0A चे फ्रेम फ़ॉमैट् (format)

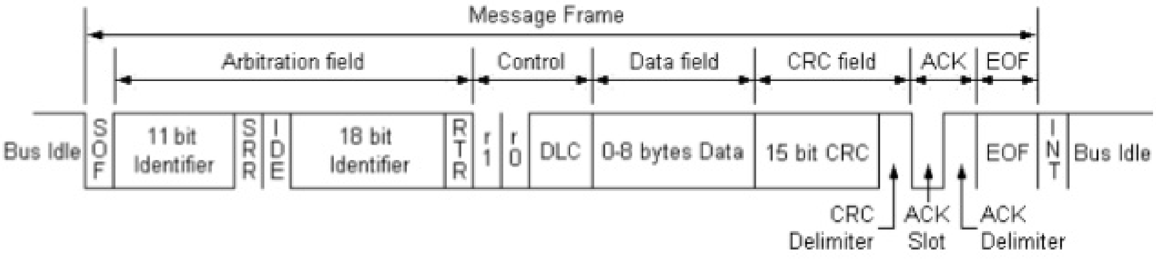


Figure.3.5: CAN 2.0B चे फ्रेम फ़ॉमैट्

Figure.3.4 आणि Figure.3.5 मध्ये दर्शविल्याप्रमाणे, विस्तारित CAN मेसेज हा स्टँडर्ड मेसेजसारखा आहे, पण त्यात खालील गोष्टींचा समावेश आहे:

a. SRR (Substitute Remote Request):

SRR बिट स्टैन्डड् मेसेजच्या RTR बिटची जागा घेते आणि विस्तारित फॉर्मॅटमध्ये प्लेसहोल्डर म्हणून कार्य करते.

b. IDE (Identifier Extension):

आयडेंटिफायर विस्तारात एक रिसेसिव्ह बिट (IDE) इन्डिकेट् करते की त्यानंतर आणखी आयडेंटिफायर बिट्स येणार आहेत. 18-बिट विस्तार IDE नंतर येतो.

c. r1:

RTR आणि r0 बिट्सनंतर, DLC बिटच्या आधी एक अडिशन्ल् रिझर्व्ह बिट समाविष्ट (included) केला आहे.

CAN बसचे फ़ीचर्:

a. डिस्ट्रीब्यूशन रिअल-टाइम कंट्रोलला समर्थन (Supports) देते, ज्यामध्ये जास्त हाय सुरक्षा पातळी असते.

b. एरर् प्रोसेस् मेकनिज़म् आणि मेसेज प्राइऑरटि कॉन्सेप्ट्ना समर्थन देते.

c. अड्वान्स्ट्(advanced) नेटवर्क रिलाइअबिलटि आणि ट्रान्समिशन इफ़िश्न्सि(efficiency).

d. मल्टी-मास्टर कपैसटिला(capacity) समर्थन (Supports) देते, जी विशेषतः (especially) नेटवर्किंग इंटेलिजेंट डिव्हायसेस, सेन्सर्स आणि अॅक्ट्युएटर्ससाठी यूस्फ़ल्(useful) असते जेव्हा ते एकाच सिस्टिम किंवा सब्-सिस्टिममध्ये जोडले जातात.

e. हाय स्पीडचे कम्यूनिकेश्न् रेट् समर्थन (Supports) करते, ज्याची स्पीड1 Mbits/sec पर्यंत असते आणि 40 फूट पर्यंत कार्य करू शकते.

f. दोन वायर, हाफ डुप्लेक्स, हाय स्पीडचे नेटवर्क सिस्टम, जे प्रामुख्याने हाय स्पीडच्या ऐप्लिकेश्न्साठी योग्य आहे.

g. ब्रॉडकास्टिंगला समर्थन करते.

h. CAN अत्यंत कठीण वातावरणात कार्य करू शकते.

CAN बसचे ऐप्लिकेश्न्:

a. इंजिन कंट्रोल युनिटसाठी, ट्रान्समिशन, एयरबॅग, अँटीलॉक ब्रेकिंग/ABS, क्रूझ कंट्रोल, इलेक्ट्रिक पावर स्टीयरिंग/EPS यांसारख्या विविध यंत्रणांसाठी.

b. ऑडिओ/व्हिडिओ सिस्टम्.

c. खिडक्या, दरवाजे, आरश्याचे ॲडजेसमेंट(Adjustment).

d. एअरक्राफ्टमध्ये फ्लाइट-स्टेट सेन्सर्स, नेव्हिगेशन सिस्टम्, फ्लाइट डेटा अनॅलसिस् ते एअरक्राफ्ट इंजिन कंट्रोल सिस्टम् जसे की इंधन सिस्टम् , पंप, आणि लिनियर ॲक्ट्युएटर्स.

e. लिफ्ट आणि एस्केलेटर.

f. स्पोर्ट्स कॅमेरे, दूरदर्शन यंत्र, स्वयंचलित दरवाजे आणि कॉफी मशीन.

3.3.3 सीरियल पेरिफेरल इंटरफेस (SPI):

सीरियल पेरिफेरल इंटरफेस (SPI) हे एक साधे 4- वायरचा सीरियल कम्युनिकेशन इंटरफेस आहे, जे अनेक मायक्रोप्रोसेसर/मायक्रोकंट्रोलर पेरिफेरल चिप्सद्वारे कंट्रोलर्स आणि पेरिफेरल डिव्हाइस् यांच्यात कम्युनिकेशन साधण्यासाठी वापरले जाते.

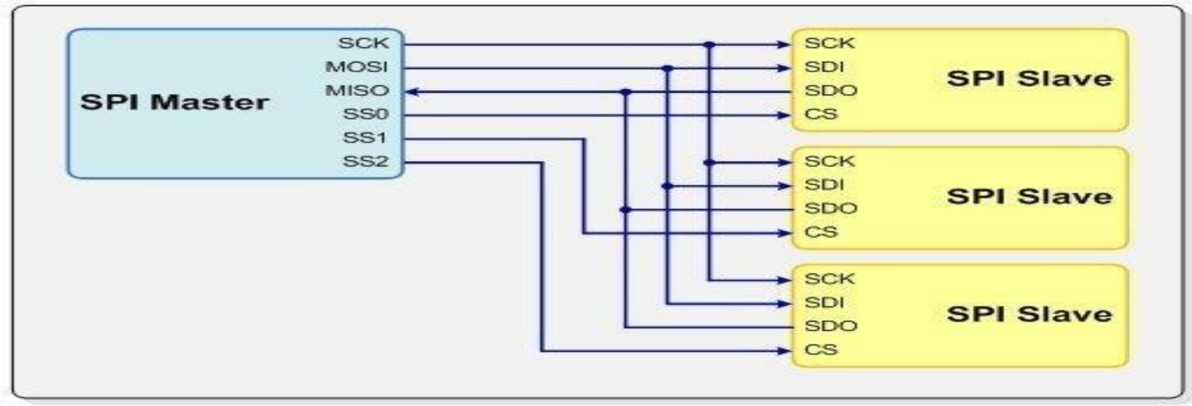


Figure.3.3.6 SPI बस मास्टर/स्लेव्ह इंटरफेस

4-वायर इंटरफेस ज्यामध्ये एक मास्टर आणि अनेक स्लेव्ह असतात, त्यात चार सिग्नल्स असतात आणि ते आहेत DATA IN, DATA OUT, CLOCK, CS (Chip Select). SPI बस एक मास्टर/स्लेव्ह इंटरफेस आहे, म्हणजे दोन डिव्हाइस् एकमेकांशी कम्युनिकेशन साधतात, ज्यामध्ये एक "मास्टर" म्हणून ओळखले जाते आणि दुसरे "स्लेव्ह डिव्हाइस" म्हणून. हे FIGURE. 3.3.6 मध्ये दर्शवले आहे.

मास्टर सिरियल क्लॉक चालवतो. SPI बस हा एक साधा सिरियल इंटरफेस आहे, जो कमीतकमी वायरच्या वापराने कमी स्पीडच्या इक्स्टऽन्ल् (external) डिव्हाइसना कनेक्ट करण्यासाठी वापरला जातो. सिरियल पेरिफेरल इंटरफेसमध्ये, डेटा एकावेळी 8 बिट्सच्या ब्लॉक्समध्ये इन किंवा आउट शिफ्ट केला जातो आणि मास्टर डिव्हाइसकडून एक किंवा अधिक स्लेव्ह डिव्हायसकडे किंवा त्यांच्याकडून डेटा ट्रान्समिट केला जातो, छोट्या अंतरावर आणि हाय स्पीडने. SPI हा फुल् ड्युप्लेक्स कम्युनिकेशन आहे, ज्यामध्ये डेटा एकाच वेळी ट्रान्समिट आणि रिसीव केला जातो.

SPI चे फ़ीचर् :

1.फुल ड्युप्लेक्स कम्युनिकेशनला समर्थन देतो.
2.हाय थ्रूपुट (throughput) प्रदान(Provide) करतो.
3.हार्डवेअर इंटरफेसिंग खूप सोपे आहे.
4.साधा(simple) 4-वायर सिरियल कम्युनिकेशन इंटरफेस.
5.डेटा ट्रान्सफर रेट् मॅक्सिमम्(maximum) 10 Mbps आहे.
6.कमी सर्किट्रीमुळे कमी पॉवर रिक्वायरमेंट
7.स्लेव्ह मास्टरचा क्लॉक वापरतो.

SPI चे ऐप्लिकेश्न्:

1.SPI विविध पेरिफेरल्सशी कम्युनिकेशन साधण्यासाठी वापरले जाते.
2.सेन्सर्स जसे की तापमान(temperature) आणि प्रेशर्(pressure), ADC, DAC, टच स्क्रीन, व्हिडिओ गेम कंट्रोलर्स.
3.ऑडिओ/व्हिडिओ कोड्स, डिजिटल पोटेन्सीमीटर.
4.कॅमेरामध्ये कॅनन EF लेन्स माउंट.
5.व्हिडिओ गेम्स.
6.फ्लॅश आणि EEPROM मेमोरी.
7.रिअल-टाइम घड्याळ.
8.LCD, LED.
9.MMC किंवा SD कार्ड, ज्यामध्ये SDIO व्हेरियंट समाविष्ट आहे.

Table3.3 :I^2C आणि CAN मधील तुलना(comprasion)

पॅरामीटर (Parameter)	**I^2C**	**CAN**
डेटा ट्रान्सफर (Data	१०० केबीपीएस, ४०० केबीपीएस आणि	२५०kbps ते १Mbps पर्यंत असिंक्रोनस.

transfer)	३.४ एमबीपीएस या तीन स्पीडसह सिंक्रोनस	
फील्डची संख्या (Number of field)	07	08 (फ्रेम एंडचे ७ बिट्स आणि इंटर फ्रेम गॅपचे ३ बिट्स समाविष्ट आहेत)
फील्डची संख्या अड्रेसिंग बिट (Number of field Addressing bit)	7- बिट or 10- बिट अड्रेस् (address)	11 बिट
ऐप्लिकेशन् (Application)	वॉच डॉग, फ्लॅश आणि रॅम मेमरी, रिअल टाइम क्लॉक, मायक्रोकंट्रोलर्स सारख्या इंटरफेस डिव्हाइससाठी	लिफ्ट कंट्रोलर्स, कॉपियर्स, टेलिस्कोप, उत्पादन-लाइन नियंत्रण सिस्टम आणि वैद्यकीय उपकरणे

ऍडवान्सड सिरियल प्रोटोकॉल्स:
वायरलेस कम्युनिकेशन म्हणजे वायरचा वापर न करता माहितीची ट्रान्समिशन. ट्रान्समिट केलेली अंतर काही मीटर असू शकते, उदाहरणार्थ, टेलिव्हिजन रिमोट कंट्रोल आणि हजारो किलोमीटर जाऊ शकते, उदाहरणार्थ, मोबाईल फोन. अधिकतर डिव्हाइस वायरलेस कम्युनिकेशनचा वापर करतात, जसे की मोबाईल्स, GPS युनिट्स, वायरलेस कॉम्प्युटर पार्ट्स आणि सॅटेलाइट कम्युनिकेशन. आता आपण काही वायरलेस कम्युनिकेशन प्रोटोकॉल्स पाहू, जसे की IrDA, Bluetooth, Zigbee.

3.4.1 IrDA (इन्फ्रारेड डेटा असोसिएशन):
परिचय(Introduction):
IrDA हे एक इन्फ्रारेड वायरलेस कम्युनिकेशन टेकनॉलजी (technology) आहे, जे इन्फ्रारेड डेटा असोसिएशनने 1993 मध्ये विकसित (developed) केले. IrDA नॉन-व्हिजिबल इन्फ्रारेड LED लाईटचा वापर एक कम्युनिकेशन मीडियम म्हणून करतो आणि वायरलेस कम्युनिकेशनसाठी हार्डवेअर तसेच सॉफ्टवेअर प्रोटोकॉल्स स्पेसिफाइ (specifies) करतो.
IrDA मध्ये डेटा ट्रान्सफरच्या सर्व लेयर्सवर(layers) कार्य करणारे प्रोटोकॉल्स असतात आणि काही नेटवर्क मॅनेजमेंट फंक्शन्स देखील असतात. IrDA हे पॉइंट-टू-पॉइंट कनेक्शन आहे, ज्याचा 30° कोन असतो. हे डेटा ट्रान्समिशन स्टॅन्डड्(standard) 0 ते 1 मीटर अंतरावर 9600 bps ते 16 Mbps च्या स्पीडने कार्य करते.
IrDA नॉन-व्हिजिबल इन्फ्रारेड लाईट स्पेक्ट्रमचा वापर करत असल्याने, कम्युनिकेशन ऑब्स्टॅकलने (obstacles)अडवले जाते, जसे की भिंती, दरवाजे, ब्रीफकेस, आणि लोक. IrDA कम्युनिकेशन कमी पॉवर वापरते आणि कमी किमतीचे आहे. IrDA हार्डवेअर देखील तुलनेने स्वस्त आहे.

IrDA चे फ़ीचर्:

अ. संपर्कापासून 1 मीटर पर्यंतचे अंतर आणि 2 मीटरपर्यंत इक्स्टेन्ड् केले जाऊ शकते.
ब. बाय-डायरेक्शनल पॉइंट-टू-पॉइंट कम्युनिकेशन.
क. 9600 b/s पासून डेटा ट्रान्समिशन, प्राइमरी स्पीड / कॉस्ट्चे स्टेप् 115 kb/s आणि मॅक्सिमम् स्पीड16 Mb/s पर्यंत.
ड. डेटा पॅकेट्स CRC वापरून संरक्षित (protected) केले जातात.
ई. कमी पॉवर आवश्यकता आणि कमी किमतीचे.
फ. अधिक सुरक्षित.

IrDA चे ऐप्लिकेश्न्:

1. PDA (Personal Digital Assistant).
2. प्रिंटर्स.
3. कॅमेरे.

4. लॅपटॉप्स आणि नोटबुक्स.
5. नोटबुक, डेस्कटॉप, आणि हँडहेल्ड कंप्युटर्स.
6. फोन आणि पेजर्स.
7. मोडेम्स.
8. LAN ॲक्सेस डिव्हाइसेस.
9. वैद्यकीय (Medical)आणि औद्योगिक (industrial) उपकरणे.
10. घड्याळे.

3.4.2 ब्लूटूथ:

परिचय (Introduction):

ब्लूटूथ हा शॉर्ट-रेंज वायरलेस कनेक्टिव्हिटी आणि डेटा ट्रान्समिशनसाठी विकसित(developed) केलेला टेकनॉलजी आहे, जो लॅपटॉप्स, कंप्युटर्स, सेल्युलर टेलिफोन्स आणि इतर इलेक्ट्रॉनिक डिव्हाइस यांच्यात कार्य करतो.
याचा मुख्य वापर वायरलेस पर्सनल एरिया नेटवर्क्स (WPAN) इस्टॅब्लिश(establish) करण्यासाठी केला जातो, ज्याला सामान्यत: ॲड-हॉक (AD-HOC) किंवा पीअर-टू-पीअर (P2P) नेटवर्क्स म्हणून ओळखले जाते. हे टेकनॉलजी आजकाल अनेक प्रकारच्या बिझनेस आणि कंझ्युमर डिव्हाइसेसमध्ये समाविष्ट केले जात आहे, जसे की मोबाइल फोन, PDA, लॅपटॉप्स, हेडसेट्स, वाहने, प्रिंटर्स.

ब्लूटूथ हे कमी किमतीचे, कमी पॉवर वापरणारे टेकनॉलजी आहे, जे लहान वायरलेस नेटवर्क प्रदान (provides) करते. या टेकनॉलजीसह डिव्हाइस एकमेकांशी शॉर्ट रेंज, ॲड-हॉक नेटवर्क्सद्वारे कनेक्ट होतात, ज्याला पिकोनेट्स म्हटले जाते. प्रत्येक वेळेस जेव्हा एक ब्लूटूथ इस्टॅब्लिश् डिव्हाइस रेडिओ प्रॉक्सिमिटीमध्ये येते किंवा बाहेर जाते, तेव्हा पिकोनेट्स ऑटोमॅटिक आणि डायनॅमिकली इस्टॅब्लिश् होतात. पिकोनेट्समधील प्रत्येक डिव्हाइस सात इतर डिव्हाइससोबत एकाच वेळी कनेक्शन अलाउ करू शकते, आणि त्या पिकोनेट्सला अनेक इतर पिकोनेट्सशी कनेक्ट होण्याची क्षमता आहे, ज्यामुळेकनेक्शनसाधतायेतात.
ब्लूटूथ मास्टर-स्लेव्ह कॉन्फिगरेशनला समर्थन देतो. कम्युनिकेशन साधण्यासाठी डिव्हाइसना एकमेकांशी पेअर किंवा कनेक्ट केले पाहिजे, आणि पेअरिंगसाठी ऑथेंटिकेशन आवश्यक आहे.

ब्लूटूथ अनलायसन्स 2.4 GHz ते 2.4835 GHz इंडस्ट्रियल, सायंटिफिक आणि मेडिकल (ISM) फ्रिक्वेंसी बँडमध्ये कार्य करते. या बँडमध्ये अनेक टेकनॉलजी , जसे की IEEE 802.11 b/g WLAN स्टॅन्डड्, कार्य करतात. ब्लूटूथ प्रत्येक ट्रान्समिशनसाठी फ्रीक्वेन्सी होपिंग स्प्रेड स्पेक्ट्रम (FHSS) वापरतो. FHSS प्रत्येक ट्रान्समिशनमध्ये ़इन्टफिअरन्स् आणि ट्रान्समिशन एररमध्ये कमी करायला मदत करते, तसेच एक मर्यादित ट्रान्समिशन सुरक्षा (security) स्तर(levels) प्रदान करते.

ब्लूटूथचे फ़ीचर् :

- 2.4 GHz वर शॉर्ट रेंज रेडिओ फ्रिक्वेन्सी.
- पॉइंट-टू-पॉइंट किंवा पॉइंट-टू-मल्टिपल पॉइंट्स प्रोटोकॉल.
- आवाज आणि डेटा ट्रान्समिशन.
- भिंतींच्या माध्यमातून 10 मीटरपर्यंत ट्रान्समिट.
- सिंक्रोनस तसेच असिंक्रोनस कम्युनिकेशनला समर्थन.

ब्लूटूथचे ऐप्लिकेशन् :

1. वायरलेस हेडसेट्स.
2. सेल फोन.
3. लॅपटॉप्स आणि नोटबुक्स.

4. PDA's (Personal Digital Assistants).
5. प्रिंटर्स.
6. वायरलेस कम्युनिकेशन (WAN).

3.4.3 झिग्बी (Zigbee):

परिचय (Introduction)

झिग्बी (ZigBee) हे एक वायरलेस कम्युनिकेशन टेकनॉलजी आहे, जे कमी किमतीच्या, कमी पॉवर वापरणाऱ्या वायरलेस M2M नेटवर्क्सच्या विशेष आवश्यकतांना संबोधित करण्यासाठी विकसित केले गेले आहे.
झिग्बी (ZigBee) IEEE 802.15.4 फिजिकल रेडिओ स्पेसिफिकेशनवर कार्य करते, जे एक पॅकेट-आधारित रेडिओ प्रोटोकॉल आहे, जे कमी किमतीच्या, बॅटरीवर चालणाऱ्या डिव्हाइससाठी डिझाइन केलेले आहे आणि हे अनलायसन्स बँड्समध्ये कार्य करते, जसे की 2.4 GHz, 900 MHz आणि 868 MHz. झिग्बी (ZigBee) डिव्हाइसना विविध नेटवर्क टॉपोलॉजींमध्ये कम्युनिकेशन साधण्याची परवानगी देतो आणि त्याची बॅटरी लाईफ अनेक वर्षांपर्यंत राहू शकते. झिग्बी (ZigBee) ही वायरलेस कम्युनिकेशन टेकनॉलजी आहे, जी रिमोट कंट्रोल आणि सेन्सर ॲप्लिकेशन्समध्ये वापरली जाते, जी कठोर(harsh) रेडिओ वातावरण आणि वेगळ्या ठिकाणी कार्य करण्यासाठी योग्य आहे.
झिग्बी (ZigBee) स्टॅन्डड् 64-बिट ॲड्रेसला तसेच 16-बिट शॉर्ट ॲड्रेसला समर्थन करते, जे प्रत्येक डिव्हाइसला युनिकपणे ओळखण्यासाठी वापरले जातात, जसे की डिव्हाइससाठी एक युनिक IP ॲड्रेस असतो. एकदा नेटवर्क सेटअप झाल्यावर, शॉर्ट ॲड्रेस वापरले जाऊ शकतात, ज्यामुळे 65000 हून अधिक नोड्सचे समर्थन केले जाऊ शकते.

झिग्बी (ZigBee) चे फ़ीचर्:

1. विविध नेटवर्क टॉपोलॉजींसाठी समर्थन, जसे की पॉइंट-टू-पॉइंट, पॉइंट-टू-मल्टिपॉइंट आणि मेष नेटवर्क्स क्लस्टर ट्री.
2. 2.4 GHz ISM बँडवर कार्य करते, जो जगभरातील बहुतेक भागांमध्ये उपलब्ध आहे.
3. कमी(low) ड्युटी सायकल(cycle) प्रदान करतो, ज्यामुळे लांब बॅटरी लाइफ मिळते.
4. 250 kbps डेटा रेट्.
5. स्केलेबल आणि सोपे(easy) डिप्लॉयमेंट.
6. उच्च रिलाइअब्ल् आणि सुरक्षितता.
7. कमी विलंब (Low latency).
8. किमतीदृष्ट्या प्रभावी. (Effective)
9. डायरेक्ट सिक्वेन्स स्प्रेड स्पेक्ट्रम (DSSS), जो कमी सिग्नल-टू-नॉईज रेशियो वातावरणात उत्कृष्ट पर्फॉरमन्स् प्रदान करतो.
10. प्रत्येक नेटवर्कमध्ये 65,000 नोड्सपर्यंत समर्थन.
11. सुरक्षित डेटा कनेक्शन्ससाठी 128-बिट AES एन्क्रिप्शन.
12. कोलिशन टाळणे, पुन्हा प्रयत्न आणि अक्नॉलेजमेंट

झिग्बी (ZigBee) चे ऍेप्लिकेश्न्:

1. मॉनिटरींग: सुरक्षा, देखरेख प्रणाली (Surveillance), आग सुरक्षा अलार्म (Fire alarms), प्रेशर् सेन्सर्स (Pressure sensors), मीटर रीडिंग मॉनिटर्स, आरोग्य आणि पर्यावरण मॉनिटर्स.
2. ऑटोमेशन आणि नियंत्रण: घर, फॅक्टरी, गोदाम, इमारत.
3. सिचूऐशन अवेरनेस आणि प्रिसिजन असेट लोकेशन (Situational awareness and precision asset location): लष्करी क्रिया, अग्निशामक ऑपरेशन्स, इन्व्हेंटरीचे रिअल-टाइम ट्रॅकिंग.
4. एंटरटेनमेंट: लर्निंग गेम, इंटरऍक्टिव्ह टॉयज.
5. कंप्युटर पेरिफेरल्स.

6. कमी पॉवर वापरणारे वायरलेस सेन्सर नेटवर्क्स.
7. सेट-टॉप बॉक्स आणि रिमोट कंट्रोल.

TABLE 3.4 IrDA आणि Bluetooth मधील तुलना(Comparison)

पॅरामीटर (Parameters)	IrDA	ब्लूटूथ (Bluetooth)
कम्युनिकेशन मीडिया	इन्फ्रारेड लाईट कम्युनिकेशन मीडिया	2.4 GHz रेडिओ फ्रिक्वेन्सी कम्युनिकेशन .
डेटा रेट	जास्तीत जास्त डेटा रेट ४ एमबी प्रति सेकंद आहे. .	जास्तीत जास्त डेटा रेट १ एमबी प्रति सेकंद आहे.
वर्किंग मोड	इन्फ्रारेड एका वेळी फक्त दोन डिव्हाइसमध्ये काम करते.	ब्लूटूथ हे जास्तीत जास्त मोबाईल डिव्हाइस काम करू शकते.
कंट्रोल सिस्टम	ओपन लूप नियंत्रण कंट्रोल सिस्टम अधिक सेनसिंटिव्ह आहे	क्लोज लूप कंट्रोल सिस्टम कमी सेनसिंटिव्ह आहे .
कम्युनिकेशन रेंज	कम्युनिकेशन रेंज १ मीटर पर्यंत आहे.	कम्युनिकेशन रेंज १00 मीटर पर्यंत आहे..
सिक्युरिटी	IrDA स्टाँग सिक्युरिटी आहे	ब्लूटूथ लेस सिक्युरिटी आहे .
कन्केशन	पॉइंट टू पॉइंट कनेक्शन ३० डिग्रीच्या नॅरो ॲगल आहे.	ओम्नी-डायरेक्शनल मल्टीपॉइंट कनेक्शन आहे
नेटवर्क	LAN (लोकल एरिया नेटवर्क) टेक्नॉलजीमध्ये IrDA उपयुक्त आहे .	ब्लूटूथ पर्सनल एरिया नेटवर्कमध्ये उपयुक्त आहे. .
कम्युनिकेशन ब्लॉक केलेले	कम्युनिकेशन ऑब्स्टॅकलमुळे अडवला जातो कारण ऑब्स्टॅकल लाईन ऑफ साईटमध्ये इन्फ्रारेड लाइटला अडवतात.	कम्युनिकेशन ऑब्स्टॅकलमुळे अडवला जात नाही कारण रेडिओ वेव्ज़चा वापर केला जातो.
अचूकता	अचूकता कमी आहे. .	अधिक अचूकता आहे .

TABLE 3.5 झिग्बी आणि ब्लूटूथ मधील तुलना (Comparison)

पॅरामीटर (Parameters)	झिग्बी	ब्लूटूथ
मॉड्युलेशन टेक्निक	DSSS (डायरेक्ट सिक्वेन्स स्प्रेड स्पेक्ट्रम) मॉड्युलेशन टेक्निकचा वापर केला जातो.	FHSS (फ्रिक्वेन्सी हॉपिंग स्प्रेड स्पेक्ट्रम) मॉड्युलेशन टेक्निकचा वापर केला जातो.
विजेचा वापर	वीज वापर कमी आहे.	जास्त वीज वापर आहे.
कम्युनिकेशन रेंज	कम्युनिकेशन रेंज 70 मीटर पर्यंत आहे.	कम्युनिकेशन रेंज रेडिओ क्लासवर अवलंबून 1 किंवा 100 मीटर आहे.
प्रोटोकॉल साइज	प्रोटोकॉल स्टॅक साइज २८ के बाइट्स आहे.	प्रोटोकॉलचा साइज २५० के बाइट्स आहे.
सेल नोड्सची संख्या	सेल नोड्सची मॅक्सिमम संख्या ६५००० पेक्षा जास्त आहे.	सेल नोड्सची मॅक्सिमम संख्या 8 आहे.
IEEE स्टँडर्ड	IEEE स्टँडर्ड 802.15.4. आहे	IEEE स्टँडर्ड 802.15.1. आहे
नेटवर्क स्पीड	मॅक्सिमम नेटवर्क स्पीड २५० बिट प्रति सेकंद आहे.	मॅक्सिमम नेटवर्क स्पीड १ एम बिट प्रति सेकंद आहे.
फ्रिक्वेन्सी बॅन्ड	फ्रिक्वेन्सी बँड २.४ GHz, ८६८ MHz आणि ९१५ MHz आहे.	फ्रिक्वेन्सी बँड २.४ GHz आहे.
टोपोलॉजीचा वापर	स्टार, ट्री, क्लस्टर ट्री आणि मेष टोपोलॉजी	ट्री टोपोलॉजी वापरली जाते.

	वापरली जातात.	

Exercise:

TLO 3.1: Compare serial vs. parallel communication and synchronous vs. asynchronous communication

1. Compare between synchronous and asynchronous communication on following parameters. (U)

a. Speed of transmission b. Form of data transmission c. Need of start and stop bit d. Cost

2. Explain Synchronous Communication with neat sketch. (U)
3. Differentiate between serial and parallel communication on following parameters. (U)

a. Data transmission speed b. Mode of transmission c. Suitable for d. Cost

4.Explain Asynchronous Communication with neat sketch. (U)

TLO 3.2: Explain the serial communication protocol.

1. Draw the pin diagram RS 232. (U)
2. State function of following pins of RS 232 (R)

a. TXD b. RXD c. CD d. DTR

3. State the function of DCE and DTE in RS232. (R)
4. List the any four features of I²C. (R)
5. Explain CAN bus protocol with neat sketch. (U)
6. List the any four features of SPI. (R)
7. Compare I²C and CAN on following parameters (U)

a. Data transfer b. Number of field c. Number of field Addressing bit d. Application

TLO 3.3: Describe the important features of advanced serial protocols.

1.List the serial and wireless communication protocols. (R)

2. List any two important features of following protocols. (R)

1. IrDA 2 Bluetooth 3. Zigbee

3. State any four features of Bluetooth Technology. (R)
4. Differentiate between IrDA and Bluetooth on following parameters. (U)

a. Accuracy b. Communication range c. Efficiency d. Technology used

5.List any four applications of ZigBee. (R)

6. List any four applications IrDA. (R)
7. List any four applications Bluetooth. (R)

8.Compare between Zigbee and Bluetooth on following parameters. (U)

a. Power consumption b. Frequency band c. Modulation techniques d. Protocol size

युनिट 4. 8051 मायक्रोकंट्रोलरसह इनपुट आणि आउटपुट डिव्हाइसेसचे इंटरफेसिंग
(Interfacing of Input and Output Devices with 8051)

Course Outcomes: Develop embedded ‘C’ programs for Input/output devices.

सिद्धांत शिक्षण निष्पत्ती (Theory Learning Outcomes)

TLO 4.1: स्विच आणि एलईडी इंटरफेसिंगसाठी 'C' प्रोग्रॅम तयार करा.
TLO 4.2: रिले इंटरफेसिंगसाठी 'C' प्रोग्रॅम तयार करा.
TLO 4.3: 7-सेगमेंट एलईडी डिस्प्ले इंटरफेसिंगसाठी 'C' प्रोग्रॅम तयार करा.
TLO 4.4: 8051 सोबत 16X2 LCD चे इंटरफेसिंग समजवा.
TLO 4.5: 8051 सोबत ADC 0808 चे इंटरफेसिंग समजवा.
TLO 4.6: स्क्रेअर वेव्ह आणि त्रिकोणी वेव्ह जनरेट करण्यासाठी 'C' प्रोग्रॅम तयार करा.

परिचय(Introduction)- 8051 मायक्रोकंट्रोलरसह इंटरफेसिंग इनपुट आणि आउटपुट (आय/ओ) डिव्हाइस एम्बेडेड सिस्टमचा एक आवश्यक पैलू आहे. इंटेलने विकसित केलेले 8051, 8-बिट मायक्रोकंट्रोलर, ऑटोमेशन, रोबोटिक्स आणि विविध नियंत्रण प्रणालीमध्ये मोठ्या प्रमाणात वापरले जाते. यात चार आय/ओ पोर्ट्स, टायमर, सिरियल कम्युनिकेशन आणि इंटरप्ट हँडलिंग क्षमता आहेत, ज्यामुळे ते रीअल-टाइम वर्सटाइल निवड आहे.

4.1 स्विच आणि LED चे इंटरफेसिंग (Interfacing of switch and LED)

स्विच (Switch): स्विच म्हणजे एक विद्युत घटक जो इलेक्ट्रॉन प्रवाह चालू किंवा बंद करण्यासाठी वापरला जातो.: चालू (closed) किंवा बंद (open).

FIGURE 4.1.1 स्विच

- **LED (Light Emitting Diode):** LED हा एक आउटपुट डिव्हाइस आहे जो सिग्नल किंवा डेटा दर्शवण्यासाठी वापरला जातो. LED विविध रंगांमध्ये उपलब्ध असतो जसे की लाल, हिरवा, पांढरा आणि पिवळा.

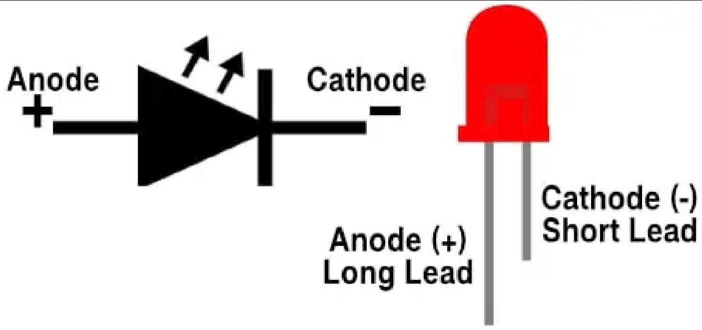


FIGURE4.1.2 LED

4.1.2 स्विच आणि LED चे 8051 सोबत इंटरफेसिंग (Interfacing of switch and LED.)

स्विच दोन स्थितींमध्ये असतो: उघडा (open) आणि बंद (closed). चालू (On) आणि बंद (Off) या स्थिती विशिष्ट व्होल्टेज स्तरांद्वारे निर्धारित केल्या जातात. स्विच योग्य हाय किंवा लो व्होल्टेज निर्माण करण्यासाठी pull-up किंवा pull-down रेसिस्टर आवश्यक असतो. Pull-up रेसिस्टर डिजिटल इनपुट आणि पुरवठा व्होल्टेज दरम्यान ठेवला जातो. जर स्विच ओपेन असेल, तर तो पोर्ट पिनला Logic 1 (high) जोडतो. जर स्विच बंद असेल, तर पोर्ट पिन Logic 0 (low) ला जोडला जातो, कारण तो ग्राउंडला जोडलेला असतो. स्विच ओपेन असल्यास, LED बंद राहतो आणि स्विच बंद झाल्यावर LED उजळतो. त्यामुळे, स्विचच्या पोर्ट पिनची स्थिती उलट करून LED च्या पोर्ट पिनला दिली जाते.

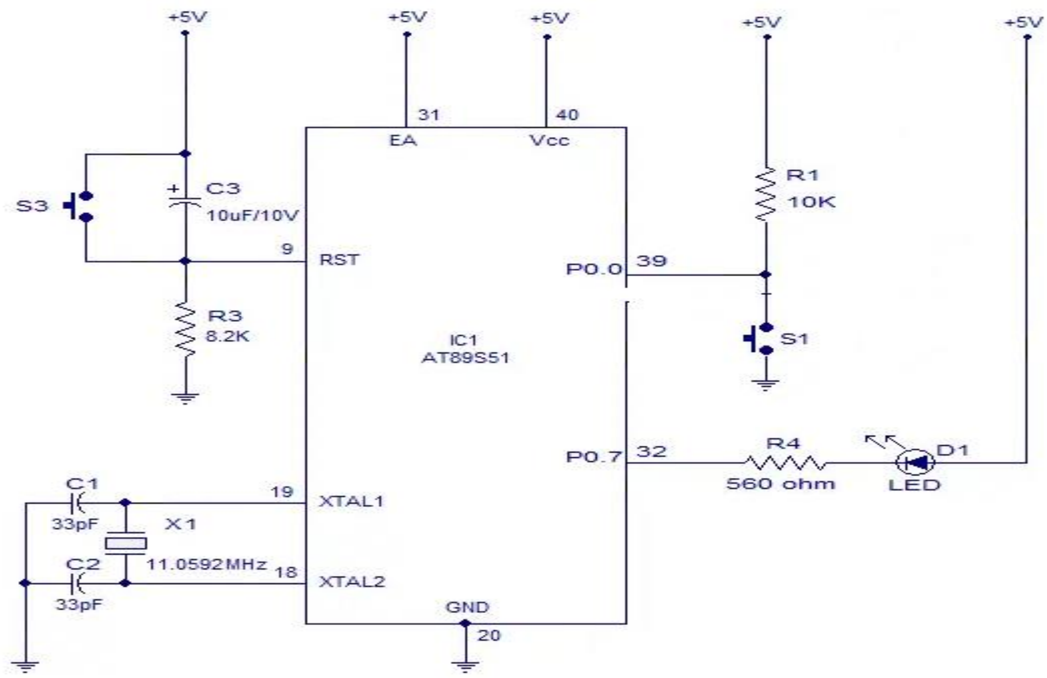


FIGURE 4.1.2.1 स्विच आणि LED चे 8051 सोबत इंटरफेसिंग

4.1.3 Algorithm

1. Initialize port pin P0.0 as input.
2. Initialize port pin P0.7 as output.
3. Read the status of key.
4. If key is closed then turn ON the LED.
5. Add some delay

6. Turn OFF the LED.
7. Repeat from step3.

4.1.4 Interfacing Program in C

```
#include <reg51.h>              // Header file for 8051 microcontroller
sbit switch_pin = P0^0          // Define switch connected to P0.0
sbit led = P0^7;                // Define LED connected to P0.7
void main()
{
switch_pin = 1;                 // ConFIGUREure switch pin as input (with pull-up)
led = 0;                        // Turn off LED initially
 while (1)
 {
   if (switch_pin == 0)         // If switch is pressed (active-low)
   {
        led = 1;                // Turn on LED
   }
   else
   {
       led = 0;         // Turn off LED
   }
 }

}
```

4.2 रिलेसोबत इंटरफेसिंग

4.2.1 रिलेचे वर्णन (Description of Relay):

रिले हे एक विद्युत, इलेक्ट्रोमेकॅनिकल किंवा इलेक्ट्रोमॅग्नेटिक स्विच आहे, जे मानवी हस्तक्षेपाशिवाय चालू (ON) आणि बंद (OFF) ऑपरेशन्स पार पाडते. म्हणूनच, ते मायक्रोकंट्रोलरच्या आउटपुट पिनला जोडले जाते आणि हाय-पॉवर डिव्हाइसेस उपकरणे जसे की मोटर, ट्रान्सफॉर्मर, हीटर, बल्ब आणि अँटेना सिस्टीम चालू/बंद करण्यासाठी वापरले जाते. रिले चालू करण्यासाठी कमी उर्जा लागते, पण ते उच्च-शक्तीची उपकरणे नियंत्रित करण्यास सक्षम असते. 8051 मायक्रोकंट्रोलरला रिले थेट जोडता येत नाही, कारण त्याला 50mA ते 70mA करंट आवश्यक असतो, जो मायक्रोकंट्रोलरच्या आउटपुट पिनद्वारे थेट पुरवला जाऊ शकत नाही. म्हणूनच, ULN2803 सारख्या ड्रायव्हर आयसीचा वापर केला जातो.

4.2.2 8051 सोबत रिलेचे इंटरफेसिंग (Interfacing of Relay with 8051):

खालील (FIGURE. 4.4) मध्ये 8051 मायक्रोकंट्रोलरसह रिले जोडण्याची पद्धत दाखवण्यात आली आहे. या इंटरफेसिंगमध्ये, ULN2803 च्या IN1 पिनला P1.0 (Port 1, Pin 0) वरून सिग्नल दिला जातो, जो रिलेच्या कार्यप्रणालीसाठी वापरला जातो.

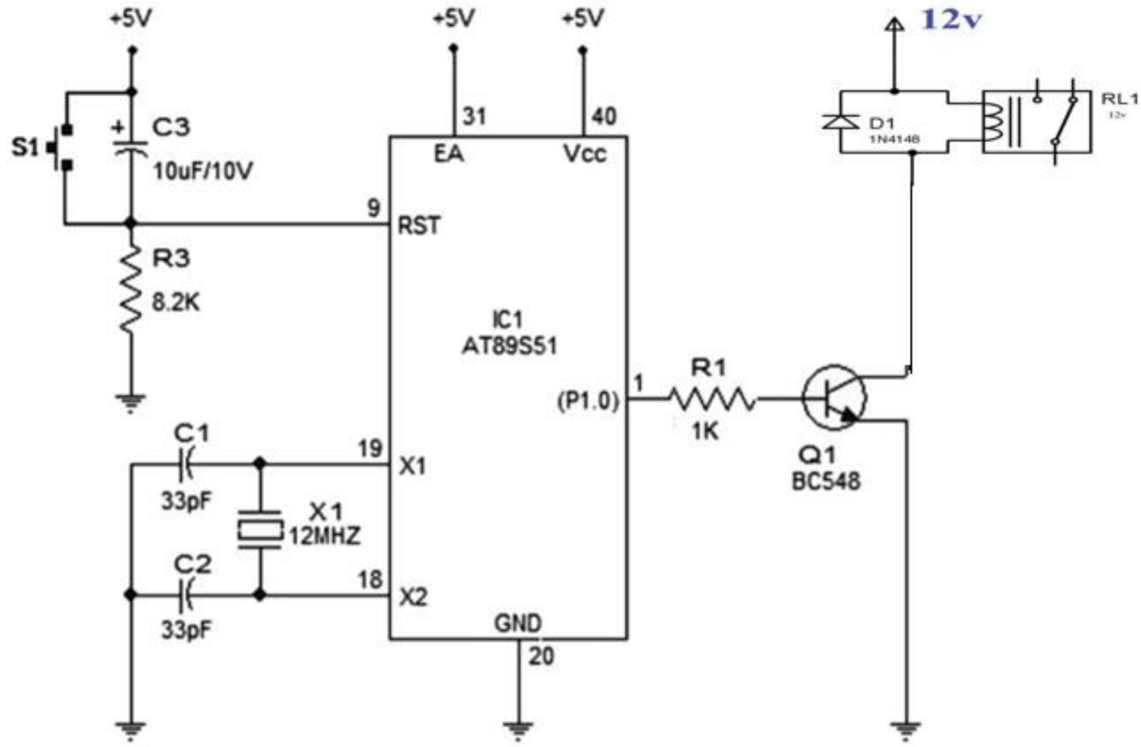


FIGURE4.2.1 8051 सोबत रिलेचे इंटरफेसिंग (Interfacing with Relay)

4.2.3 Algorithm

1. Initialize relay to port pin P1.0.
3. Initially relay turned OFF.
4. Turn ON relay.
5. Add some delay
6. Turn OFF relay.
7. Add some delay
8. Repeat from step4.

4.2.4 Interfacing Program in C

```
#include <reg51.h>
sbit relay = P1^0;            // Define relay connected to P1.0
void delay()
{
  int i, j;
  for (i = 0; i < 500; i++)
  {
  for (j = 0; j < 1000; j++);
  }
}
void main()
{
  relay = 0;                  // Initially, relay is OFF
  while (1)
```

```
    {
     relay = 1;                  // Turn ON relay
     delay();                    // Wait for some time
     relay = 0;                  // Turn OFF relay
      delay ();                  // Wait for some time
      }
  }
```

4.3: 7-सेगमेंट LED डिस्प्लेसोबत इंटरफेसिंग

4.3.1: 7-सेगमेंट डिस्प्लेचे वर्णन:

7-सेगमेंट डिस्प्ले औद्योगिक क्षेत्रात मोठ्या प्रमाणावर वापरले जाते आणि ते 0 ते 9 अंक तसेच A, B, C, D यांसारखी काही अक्षरे दर्शवू शकते. 7-सेगमेंट डिस्प्लेचे दोन प्रमुख प्रकार असतात:

- कॉमन कॅथोड (Common Cathode) – सर्व कॅथोड पिन ग्राउंडला जोडलेले असतात.
- कॉमन अ‍ॅनोड (Common Anode) – सर्व अ‍ॅनोड पिन हाय व्होल्टेजला (Vcc) जोडलेले असतात.

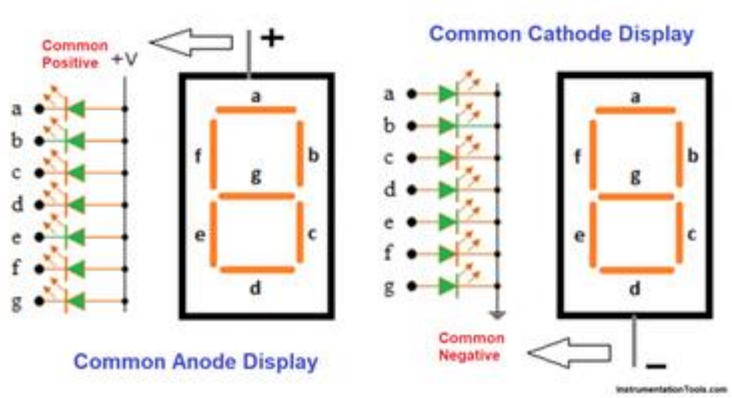


FIGURE4.3.1 कॉमन कॅथोड , कॉमन अ‍ॅनोड

सर्व एलईडी विभागांचे कॅथोड एकाच टर्मिनलशी एकत्र जोडलेले आहेत ज्याला 'COM' असे लेबल लावले जाते आणि सर्व एलईडी विभागातील एनोड ए, बी, सी, डी, ई, एफ, जी असे लेबल केलेले पिन म्हणून ओपेन ठेवले आहेत.

FIGURE.4.3.2 7-सेगमेंट डिस्प्ले

7 सेगमेंट डिस्प्लेचे सामान्य अंक ड्राइव्ह नमुने (0 ते 9) TABLE 4.3.1 मध्ये दर्शविले आहेत

TABLE 4.3.1 Seven Segment Display Code

Numbers	Common Cathode		Common Anode	
	(DP)GFEDCBA	HEX Code	(DP)GFEDCBA	HEX Code
0	00111111	0x3F	11000000	0xC0
1	00000110	0x06	11111001	0xF9
2	01011011	0x5B	10100100	0xA4
3	01001111	0x4F	10110000	0xB0
4	01100110	0x66	10011001	0x99
5	01101101	0x6D	10010010	0x92
6	011111101	0x7D	10000010	0x82
7	00000111	0x07	11111000	0xF8
8	01111111	0x7F	10000000	0x80
9	01101111	0x6F	10010000	0x90

4.3.1 8051- सह 7-सेगमेंट डिस्प्लेचे इंटरफेसिंग (Interfacing of a 7-segment Display with 8051)-

4.3.2 FIGURE4.3.3 8051 मायक्रोकंट्रोलरसह 7 सेगमेंट एलईडी डिस्प्लेचे इंटरफेसिंग दर्शविते.

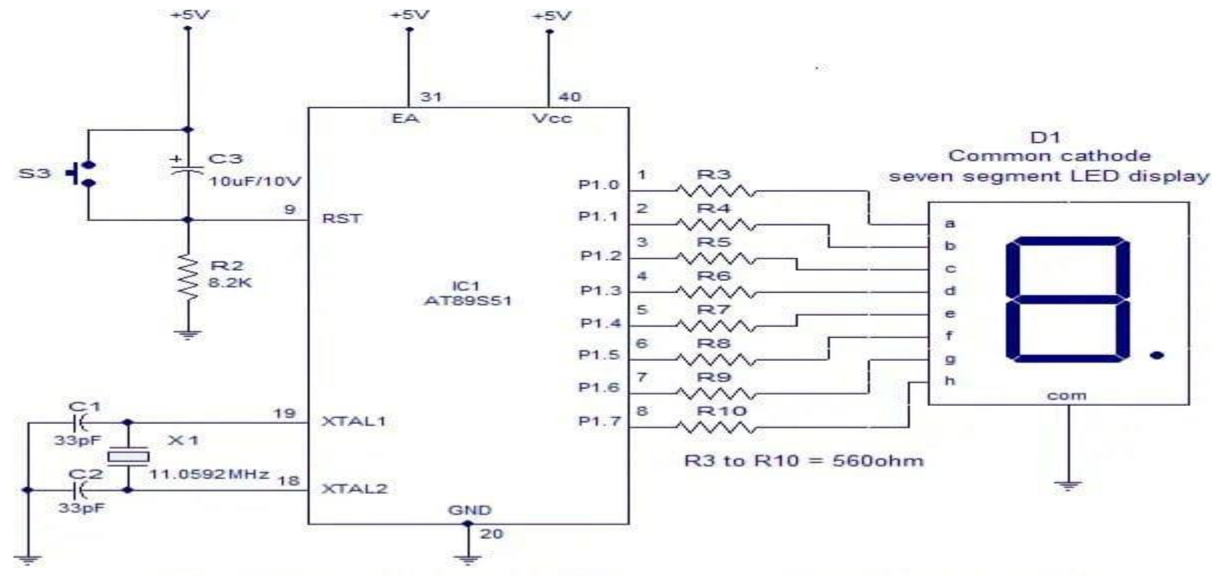


FIGURE4.3.3 मायक्रोकंट्रोलरसह 7 सेगमेंट एलईडी डिस्प्लेचे इंटरफेसिंग

4.3.3 Algorithm

1. Make port P1 as output
2. Initialize all the segment hex values of the digits in an array
example:
unsigned char arr[10]={ 0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x7D,0x07,0x7F,0x6F};
Take for loop and assign array values to the PORTI
3. Add time delay
4. Repeat the process from step 2.

4.3.3 Interfacing Program in C

```
#include <reg51.h>
unsigned char digits [10] = {
    0x3F, // 0
    0x06, // 1
    0x5B, // 2
    0x4F, // 3
    0x66, // 4
    0x6D, // 5
    0x7D, // 6
    0x07, // 7
    0x7F, // 8
    0x6F  // 9
};

// Delay function
    void delay(unsigned int ms)
 {
     unsigned int i, j;
    for (i = 0; i < ms; i++)
    for (j = 0; j < 1275; j++);                         // Approximate delay
 }

    void main()
 {
    unsigned char i;
    while (1)
     {
     for (i = 0; i < 10; i++)
     {
           P2 = digits[i]; // Send data to 7-segment
           delay(1000);    // 1-second delay
      }
    }
}
```

4.4: 8051 सह 16x2 एलसीडीचे इंटरफेसिंग

4.4.1 16x2 एलसीडीचे वर्णन (Description of 16X2 LCD) :-

एलसीडी हा एक अल्फान्यूमेरिक डिस्प्ले आहे, जो विविध आकारात उपलब्ध आहे एलसीडीची निवड सामान्यत: 16x2 एलसीडी व्यापकपणे वापरली जाते 16x2 एलसीडी म्हणजे 16 कॅरॅक्टर्स पर लाइन आणि तेथे 2 डेटा रजिस्टर आणि कमांड रजिस्टर असतात. प्रदर्शित करण्यासाठी डेटा रजिस्टरकडे डेटा आहे कमांड रजिस्टरमध्ये आदेश साठवले जातात, जे डिस्प्ले नियंत्रित करतात.

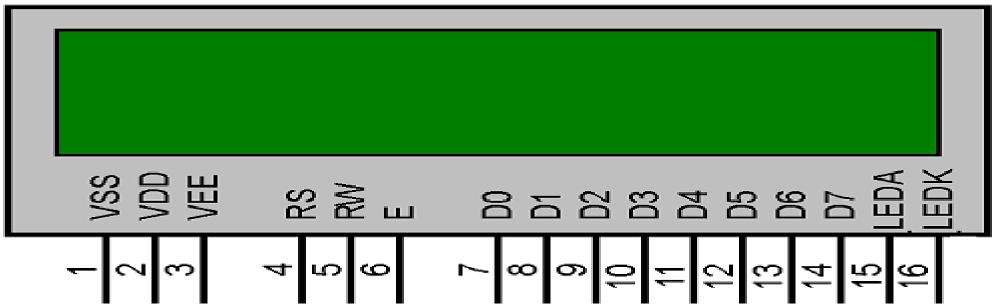


FIGURE4.4. एलसीडीचे पिन

4.4.2 एलसीडी पिनचे कार्य (Function of LCD pins) –

1) व्हीएसएस, व्हीडीपी आणि व्हीई - व्हीएसएस = 5 व्ही वीजपुरवठा, व्हीडीडी = ग्राउंड, व्हीई = एलसीडी कॉन्ट्रास्ट नियंत्रित करीत आहे.

2) डीओ-डी 7-8 बिट डेटा पिन. डीओ-डी 7 एलसीडीला माहिती पाठविण्यासाठी किंवा एलसीडी इंटर्नल रजिस्टरस् read करण्यासाठी.

3) आरएस-रजिस्टर सिलेक्ट-आरएस एकतर डेटा रजिस्टर किंवा कमांड रजिस्टर निवडण्यासाठी वापरलेला रजिस्टर सिलेक्ट पिन आहे.

आरएस = 1 = डेटा रजिस्टर, आरएस = 0 = कमांड रजिस्टर

4) आर/डब्ल्यू- हा पिन रीड अँड राइट करण्यासाठी वापरला जातो.

आर/डब्ल्यू 1 = वाचन = वाचन डेटा, आर/डब्ल्यू = 0 = लिहा = एलसीडीला डेटा लिहिणे

5) इनएबल: इनएबल पिन मॉड्यूल प्रारंभ करण्यासाठी आहे. निवडलेल्या रजिस्टरमध्ये लॅचिंग डेटा किंवा कमांड पिनसाठी हाय टू लो पल्स दिली जाते.

4.4.3 एलसीडी कमांड

TABLE 4.4.1 LCD Commands

No	HEX Value	COMMAND TO LCD
1	0x01	Clear Display Screen
2	0x30	Function Set: 8-bit, 1 Line, 5x7 Dots
3	0x38	Function Set: 8-bit, 2 Line, 5x7 Dots
4	0x20	Function Set: 4-bit, 1 Line, 5x7 Dots
5	0x28	Function Set: 4-bit, 2 Line, 5x7 Dots
6	0x06	Entry Mode
7	0x08	Display off, Cursor off
8	0x0E	Display on, Cursor on
9	0x0C	Display on, Cursor off
10	0x0F	Display on, Cursor blinking
11	0x18	Shift entire display left
12	0x1C	Shift entire display right
13	0x10	Move cursor left by one character
14	0x14	Move cursor right by one character
15	0x80	Force cursor to beginning of 1st row
16	0xC0	Force cursor to beginning of 2nd row

4.3.4 8051 सह 16 x 2 एलसीडीचे इंटरफेसिंग (Interfacing of 16 X2 LCD with 8051)

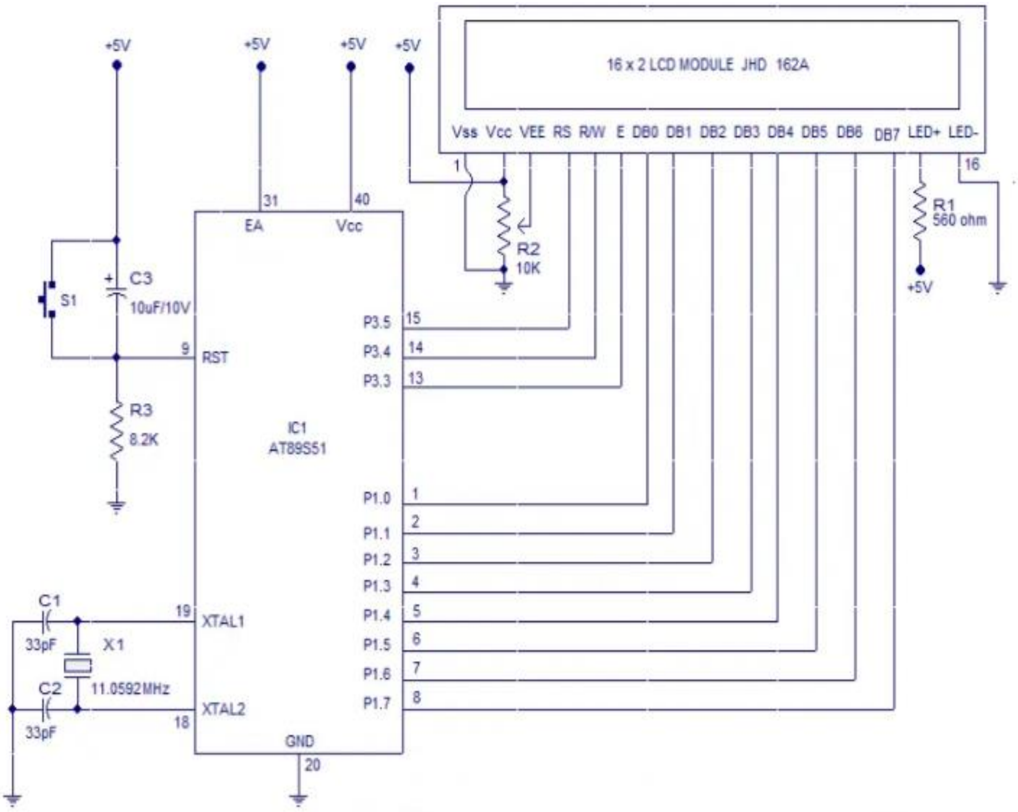


FIGURE4.4.2 8051 सह 16 x 2 एलसीडीचे इंटरफेसिंग

4.3.2 Algorithm

1. Initialize the LCD using the LCD initialization commands.
2. Set the address of DDRAM, where you want to display the character.
3. Write the character on the data bus of LCD.
4. If passing the command then make RS pin low either for data make RS (RS = 0) pin high (RS - 1).
5. Set Enable high (EN - 1)
6. Write the command the data bus.
7. Set Enable pin low (EN = 0)

4.3.3 Interfacing program of 16 X2 LCD with 89C51

```
#include <reg51.h>
#define LCD P2  // Connect LCD data pins to Port 2
sbit RS = P3^5; // Register Select pin
sbit RW = P3^4; // Read/Write pin
sbit EN = P3^3; // Enable pin
void delay(unsigned int time)
 {
    unsigned int i, j;
    for (i = 0; i < time; i++)
       for (j = 0; j < 1275; j++);
}
```

```
    void lcd_cmd(unsigned char command)
{
    LCD = command;
    RS = 0; // Command mode
    RW = 0; // Write mode
    EN = 1;
    delay(1);
    EN = 0;
    delay(2);
}
    void lcd_data(unsigned char data)
{
    LCD = data;
    RS = 1; // Data mode
    RW = 0; // Write mode
    EN = 1;
    delay(1);
    EN = 0;
    delay(2);
}
    void lcd_init() {
    lcd_cmd(0x38);              // 8-bit mode, 2 lines, 5x7 font
    lcd_cmd(0x0C);             // Display ON, Cursor OFF
    lcd_cmd(0x06);             // Auto-increment cursor
    lcd_cmd(0x01);            // Clear display
    delay(5);
}
    void lcd_string(char *str)
{
    while (*str)
    {
    lcd_data(*str);
    str++;
    }
}
   void main()
{
    lcd_init();
    lcd_string("Hello, 8051!");
    while (1);
}
```

4.5 8051सह एडीसी 0808 चे इंटरफेसिंग (Interfacing of ADC 0808 with 8051.)

4.5.1 Description to ADC: एम्बेडेड सिस्टममध्ये, असे बरेच सिग्नल आहेत जे निसर्गात अॅनालॉग आहेत, परंतु मायक्रोकंट्रोलर किंवा एम्बेडेड प्रोसेसर एक डिजिटल डिव्हाइस आहे. म्हणूनच, आम्हाला एक डिव्हाइस आवश्यक आहे जे एडीसी (एनालॉग टू डिजिटल कन्व्हर्टर) म्हणून ओळखल्या जाणाऱ्या डिजिटल आउटपुटमध्ये एनालॉग इनपुटचे भाषांतर करते. नियंत्रकासह एडीसीला इंटरफेस करण्यापूर्वी आम्ही प्रथम डिजिटल रूपांतरणात अॅनालॉगच्या मूलभूत गोष्टींकडे पाहतो.

4.5.2 एडीसी 0808/0809 ची वैशिष्ट्ये (Specifications of ADC 0808/0809):

1. 8-बिट क्रमिक अनुमान एडीसी.
2. 8-चॅनेल मल्टिप्लेक्सर. एड्रेस लॉजिकसह
3. परिवर्तन कालावधी(conversion time) 100 µs.
4. बाह्य शून्य व पूर्ण-स्केल समायोजनाची आवश्यकता निष्प्रभ करते
5. सर्व मायक्रोप्रोसेसरशी इंटरफेस करणे सोपे आहे.
6. आउटपुट टीटीएल व्होल्टेज स्तरावरील वैशिष्ट्ये पूर्ण करा.
7. 0 व्होल्ट ते 5 व्होल्ट इनपुट रेन्ज

4.5.3 एडीसी 0808/0809 चे पिन वर्णन (Pin description of ADC 0808/0809):

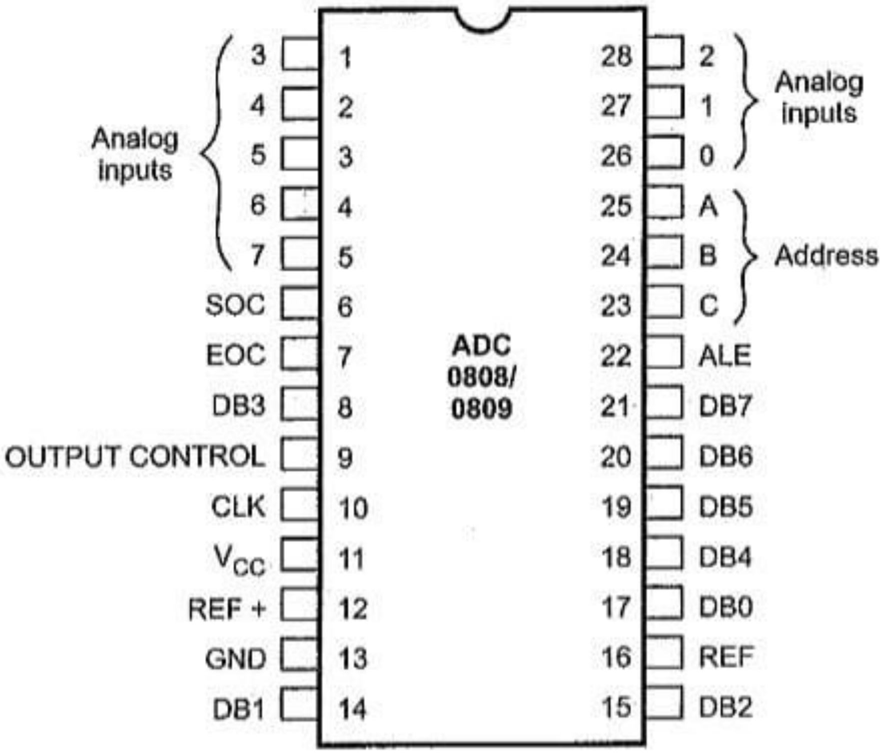


FIGURE4.5.1 एडीसी 0808 पिन डायग्रॅम

4.5.4 ADC0808 मधून डेटा अॅक्सेस करण्याची स्टेप्स (Steps to access data from ADC0808)

1. अॅनालॉग चॅनेल निवडा:A, B आणि C अॅड्रेस पिन्सला योग्य बिट्स प्रदान करून योग्य अॅनालॉग चॅनेल निवडा (तक्त्यानुसार).
2. ALE (Address Latch Enable) सक्रिय करा:ALE पिनला L-to-H पल्स द्या जेणेकरून निवडलेला पत्ता लॅच होईल.
3. SC (Start Conversion) सक्रिय करा:SC पिनला L-to-H पल्स द्या, जेणेकरून कन्व्हर्जन प्रक्रिया सुरू होईल.

4. EOC (End of Conversion) मॉनिटर करा:EOC पिन तपासा की कन्व्हर्जन पूर्ण झाले आहे का.,जर EOC H-to-L बदलला तर कन्व्हर्जन पूर्ण झाले आहे आणि डेटा वाचण्यासाठी तयार आहे.
5. OE (Output Enable) सक्रिय करा आणि डेटा वाचा:OE पिनला L-to-H पल्स द्या, त्यामुळे डिजिटल डेटा ADC चिपमधून बाहेर येईल.,लक्षात घ्या की OE हा इतर ADC चिपमध्ये RD (Read) पिनसारखाच कार्य करतो.

TABLE 4.5.1 Channel Selection

Select Analog Channel	C	B	A
IN0	0	0	0
IN1	0	0	1
IN2	0	1	0
IN3	0	1	1
IN4	1	0	0
IN5	1	0	1
IN6	1	1	0
IN7	1	1	1

4.5.4 Algorithm

1. Select an analog channel by providing bits to A, B, and C addresses according to the analog signal selection.
2. Activate the ALE (address latch enable) pin.
3. Activate SC (start conversion) to initiate conversion.
4 Monitor EOC (end of conversion) to see whether conversion is finished. H-40 L output indicates that the data is converted and is ready to be picked up. If we do not use EOC, we can read the converted digital data after a brief time delay. The delay size depends on the speed of the external clock we connect to the CLK pin.
5. Activate OF (output enable) to read data out of the ADC chip

4.5.5 Interfacing of ADC0808

FIGURE4.5.2 shows interfacing of ADC with 8051 in which port 1is used to interface data lines D0 to D7.

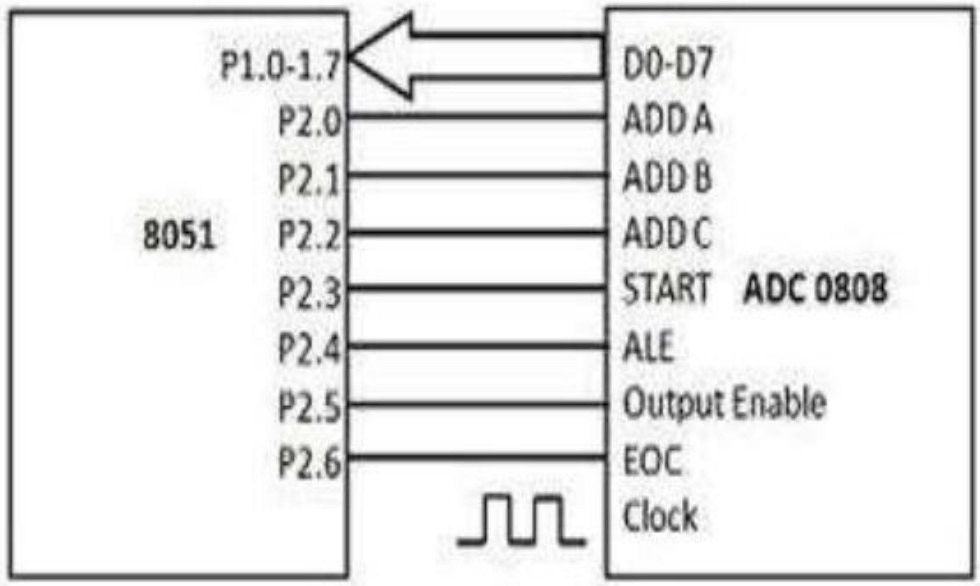


FIGURE 4.5.2 एडीसीचे इंटरफेसिंग

4.5.5 Interfacing Program

```
#include <reg51.h>
// Define control pins
sbit ALE = P2^4;                    // Address Latch Enable
sbit SC = P2^3;                     // Start Conversion
sbit EOC = P2^6;                    // End of Conversion
sbit OE = P2^5;                     // Output Enable

// Define address lines for channel selection
sbit A = P2^0;
sbit B = P2^1;
sbit C = P2^2;

void delay()
{
   int i;
   for(i = 0; i < 3000; i++);          // Simple delay loop
}

unsigned char ADC_Read(unsigned char channel)
{
```

```
    unsigned char result;
    A = (channel & 0x01);
    B = (channel & 0x02) >> 1;
    C = (channel & 0x04) >> 2;

    // Latch the address
    ALE = 1;
    delay();
    ALE = 0;

    // Start conversion
    SC = 1;
    delay();
    SC = 0;

    // Wait for conversion to complete
    while (EOC == 1);  // Wait until EOC goes LOW
    while (EOC == 0);  // Wait until EOC goes HIGH
    // Enable output and read data
    OE = 1;
    result = P2;                          // Read 8-bit digital output from ADC
    OE = 0;
    return result;
}
void main()
{
    unsigned char adc_value;
    P2 = 0xFF;                            // ConFIGUREure Port 2 as input
    while (1)
    {
        adc_value = ADC_Read(0);          // Read from channel 0
    }
}
```

4.6: स्क्वेअर वेव्ह आणि ट्रायँग्युलर वेव्ह जनरेट करण्यासाठी ‘C’ प्रोग्राम

4.6.1 डीएसीचे वर्णन (Description of DAC):

डिजिटल टू एनालॉग कन्व्हर्टर (डीएसी) एक डिव्हाइस आहे जे डिजिटल डेटाला एनालॉग सिग्नलमध्ये रूपांतरित करते. एक डीएसी अचूकतेसह एनालॉग सिग्नलमध्ये नमुना घेतलेला डेटा पुन्हा तयार करू शकतो. डिजिटल डेटा मायक्रोप्रोसेसर, विशिष्ट इंटिग्रेटेड सर्किट (एएसआयसी) किंवा फील्ड प्रोग्राम करण्यायोग्य गेट संचयिका (एफपीजीए) द्वारे दिला जाऊ शकतो, परंतु शेवटी संवाद साधण्यासाठी डेटाला अॅनालॉग सिग्नलमध्ये रूपांतरण आवश्यक आहे.

4.6.2 डीएसी 0808 ची वैशिष्ट्ये (Specifications of DAC 0808):

1. 8 बिट समांतर डिजिटल डेटा इनपुट
2. त्वरित सेटलमेंट वेळ (ठराविक मूल्य): 150 ns.
3. सापेक्ष अचूकता ±0.19% कमाल त्रुटीसह

4. पूर्ण-स्केल प्रवाह जुळवणी: ±1 LSB

5. नॉन-इन्व्हर्टिंग डिजिटल इनपुट्स TTL आणि CMOS सुसंगत आहेत, तसेच चॅनेलचा पत्ता (Address) समर्थित आहे.

6. हाय स्पीड गुणाकार इनपुट स्लू रेट: 8 mA/us

4.6.2 डीएसी 0808 चे पिन वर्णन(Pin Description of DAC 0808):

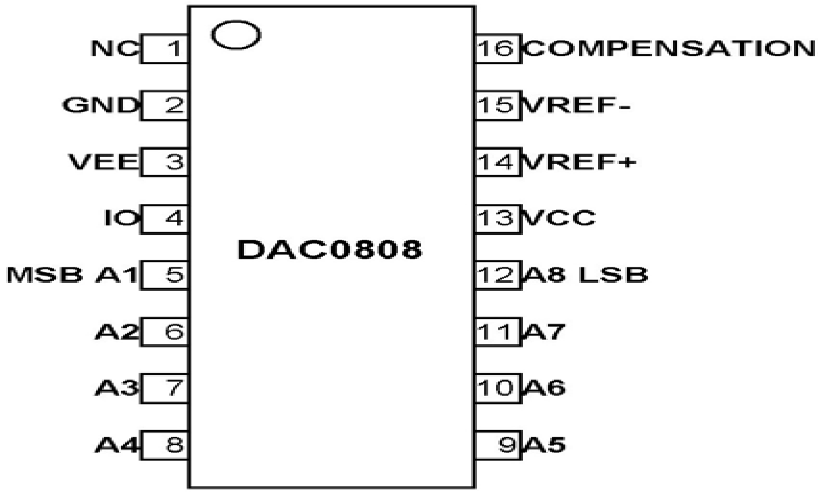


FIGURE4.6.1 पिन डायग्रॅम

डीएसी 0808 चे पिन वर्णन TABLE 4.6.१ मध्ये दिले आहे

TABLE 4.6.1 पिन वर्णन

Pin No	Name	Description
1	NC	No connection
2	GND	Ground
3	VEE	Negative power supply
4	A1 to A8	Digital inputs
5	VCC	Positive power supply
6	VREF+	Positive reference voltage
7	VREF-	Negative reference voltage
8	COMP	Compensation capacitor pin

4.6.3 डीएसीचे इंटरफेसिंग

FIGURE 4.6.2 मध्ये 8051 मायक्रोकंट्रोलरसह डीएसी 0808 चे इंटरफेसिंग दर्शविले गेले आहे, जेथे पोर्ट 0 डीएसी 0808 च्या डेटा पिनशी जोडलेले आहे म्हणजे A1 ते A8. डीएसीचे आउटपुट विद्युत प्रवाह स्वरूपात आहे, परंतु आउटपुट म्हणून व्होल्टेजची आवश्यकता आहे, तर विद्युत प्रवाह ते व्होल्टेज कन्व्हर्टेड वापरले गेले आहे.

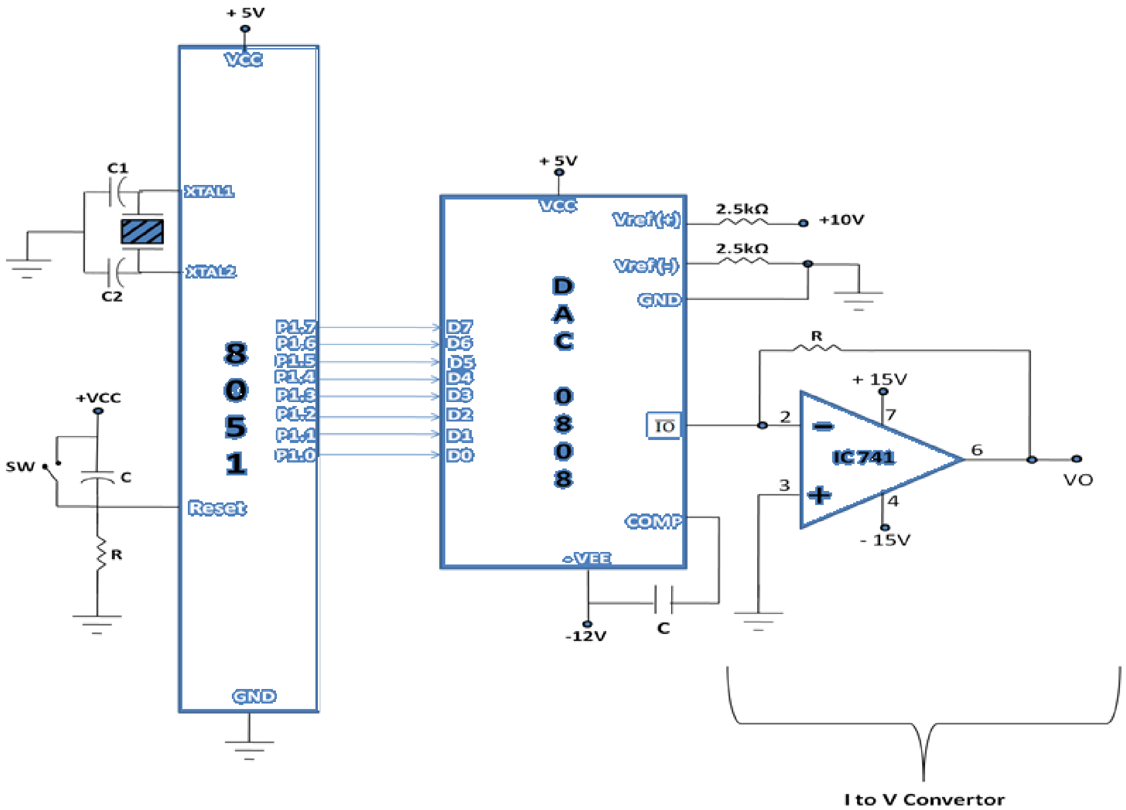


FIGURE 4.6.2 डीएसीचे इंटरफेसिंग

4.6.3 अल्गोरिदम (Algorithm)

1. Initialize port Pl as output port.
2. Send maximum value at port P1.
3. Call delay for calculated time interval to generate square wave i.e time delay.
4. Send minimum value at port P1.
5. Call same delay to obtain 50% duty cycle.
6. Repeat from step2.

4.6.4 ‘C’ program to generate square wave

```
#include<reg51.h>
void  delay(void);
void main(void)
{
while(1)
{
  P1 = 0x00;                    // Min output (Low level)
      delay();
         P1=0xff;               // Max output (High level)
delay();
}
}
void delay(void)
```

```
{
unsigned int i, j;
for(i=0;i<5000;i++)
for(j=0;j<5000;j++);
}
```

4.6.3 ‘C’ program to generate triangular wave

```
#include<reg51.h>
void main(void)
{
unsigned char d;
  while (1)
  {
     for (d=0; d<255; d++)       // Rising edge of the triangular wave
    {
     P1 = d;                      // Output increasing values to DAC
    }
     for (d=255; d>0; d--)       // Falling edge of the triangular wave
   {
    P1 = d;                       // Output decreasing values to DAC
    }
  }
}
```

Exercise-

TLO 4.1: Develop ‘C’ program for interfacing switch and LED.

1. Write a C language program to turn ON LED after pressing a switch. (A)
2. Write 8051 'C' program turn ON and OFF LED after some delay with labeled interface diagram. (A)
3. Sketch the interfacing diagram of LED and Switch with 8051 microcontroller. (A)

TLO 4.2: Develop ‘C’ program for interfacing a relay.

1. Draw the labelled diagram to interface relay with 8051 microcontroller. (A)
2. Write C language program to make relay ON/OFF after certain delay with neat diagram of interfacing relay with 8051 microcontroller. (A)

TLO 4.3: Develop ‘C’ program for interfacing 7-segment LED display.

1. Draw the interfacing of seven segment display with 8051 microcontroller. (A)
2. Interface 7-segment display to Port 2 of microcontroller 8051 and write a program to display number ‘2’ on it.(A)

TLO 4.4: Explain interfacing of 16X2 LCD with 8051.

1. Draw the pin diagram of 16X2 LCD display. (U)
2. State the function of following pins. (R)
   1) RS 2) EN 3) R/W
3. Draw the interfacing of LCD display to 8051 microcontroller. (A)
4. Write a C program to display ‘WELCOME’ on 16 × 2 LCD display. (A)
5. State 16X2 LCD commands with its functions. (Any 4)(R)

TLO 4.5: Explain interfacing of ADC 0808 with 8051.

1. Draw labeled diagram to interface ADC 0808 with microcontroller 8051. (A)
2. State the specifications of ADC 0808. (R)
3. Write the C program to interface ADC 0808 with 8051 microcontroller. (A)

TLO 4.6: Develop 'C' program for generating square wave and triangular wave.

1 List the specifications of DAC 0808. (R)
2. Write C language program to generate square wave with neat diagram to interface DAC 0808 with 8051. (A)
3. Write C language program to generate triangular wave with neat diagram to interface DAC 0808 with 8051. (A)
4. Draw the pin diagram of DAC0808. (U)
5. Sketch the diagram to interface DAC 0808 to 8051 microcontroller. (U)

युनिट 5 8051 मायक्रोकंट्रोलरचे एप्लीकेशन

(Applications of 8051 Microcontroller)

Course Outcomes: Develop basic embedded system applications.

सिद्धांत शिक्षण निष्पत्ती (Theory Learning Outcomes)

TLO 5.1: 8051 सोबत इंटरफेस केलेल्या DC मोटर आणि सर्व्हो मोटर रोटेट करण्यासाठी 'C' प्रोग्रॅम तयार करा.
TLO 5.2: स्टेपर मोटरला डावीकडून उजवीकडे आणि उजवीकडून डावीकडे फिरवण्यासाठी 'C' प्रोग्रॅम तयार करा.
TLO 5.3: IR सेन्सर वापरून अडथळा शोधण्यासाठी 'C' प्रोग्रॅम तयार करा.
TLO 5.4: PIR सेन्सर वापरून हालचाल शोधण्यासाठी 'C' प्रोग्रॅम तयार करा.

5.1 **डीसी मोटर** -डीसी मोटर्सचे अनेक उपयोग आहेत, ज्यात इलेक्ट्रिक वाहने, रोबोटिक्स आणि अक्षय ऊर्जा यांचा समावेश आहे. डीसी मोटर्स डायरेक्ट करंट (डीसी) द्वारे चालवल्या जातात.

FIGURE 5.1 डीसी मोटर

डीसी मोटर डायरेक्ट करंटच्या स्वरूपात असलेल्या विद्युत उर्जेचे यांत्रिक उर्जेमध्ये रूपांतर करते. मोटरच्या बाबतीत, निर्माण होणारी यांत्रिक ऊर्जा मोटर शाफ्टच्या फिरण्याच्या हालचालीच्या स्वरूपात असते. मोटरमधून डायरेक्ट करंटची दिशा उलट करून मोटरच्या शाफ्टच्या फिरण्याची दिशा उलट करता येते. मोटरला एक निश्चित व्होल्टेज देऊन विशिष्ट वेगाने फिरवता येते. जर व्होल्टेज बदलत असेल तर मोटरचा वेग बदलतो. अशा प्रकारे, डीसी मोटरचा वेग वेगवेगळा डीसी व्होल्टेज लावून नियंत्रित करता येतो; तर मोटरमधून येणाऱ्या विद्युत प्रवाहाची दिशा उलट करून त्याच्या फिरण्याची दिशा बदलता येते. Table 5.1 नियंत्रण सिग्नल आणि मोटरची स्थिती दर्शविते.

Table 5.1 इनपुट सिग्नल आणि मोटर स्थिती

इनपुट 1	इनपुट 2	मोटार स्थिती
Low	Low	मोटार थाबंते
Low	High	मोटारघड्याळाच्या दिशेने फिरते
High	Low	मोटारघड्याळाच्या उलट दिशेने फिरते
High	High	मोटार थाबंते

5.1.1 **डीसी मोटर ड्रायव्हर सर्किट** -AT89C51 मायक्रोकंट्रोलर (8051 कुटुंबातील) वापरून DC मोटर नियंत्रित करता येते. तथापि, मायक्रोकंट्रोलर मोटर चालविण्यासाठी आवश्यक असलेला करंट आणि व्होल्टेज थेट प्रदान करू शकत नसल्यामुळे, H-ब्रिज ड्रायव्हर (L293D) वापरला जातो. वेगवेगळे व्होल्टेज लागू करण्यासाठी, आपण PWM तंत्राचा वापर करू शकतो. करंट रिव्हर्स करण्यासाठी, आपण H-ब्रिज सर्किट किंवा मोटर ड्रायव्हर IC चा वापर करू शकतो जे H-ब्रिज तंत्र किंवा इतर कोणत्याही यंत्रणेचा वापर करतात. FIGURE5.1.1 L293D साठी पिन आउट FIGURE दर्शविते.

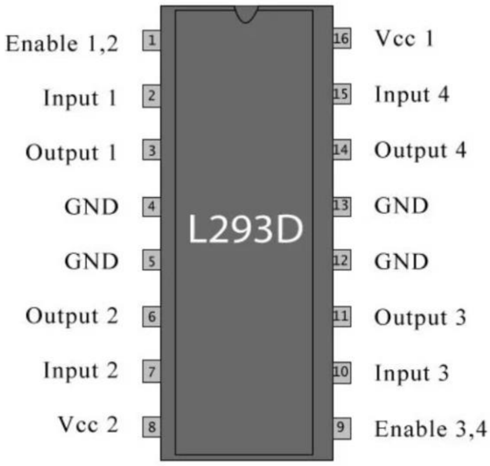


FIGURE 5.1.1 L293D चा पिन आउट

5.1.2 **89C51 मायक्रोकंट्रोलरसह डीसी मोटरचे इंटरफेसिंग** -FIGUREure 5.1.1 मध्ये L293D मोटर ड्रायव्हर IC वापरून 8051 मायक्रोकंट्रोलरसह DC मोटरचे इंटरफेसिंग दाखवले आहे. पोर्ट P1 चे दोन पोर्ट पिन P1.0 आणि P1.1 L293D शी जोडलेले आहेत.

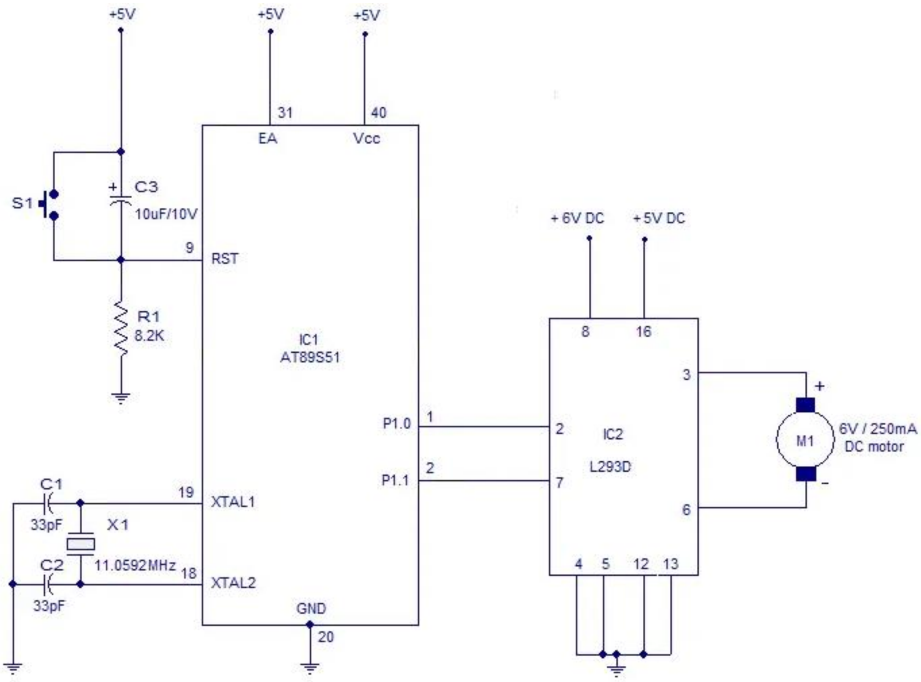


FIGURE5.1.2 L293D वापरून 80C51 मायक्रोकंट्रोलरसह डीसी मोटरचे इंटरफेसिंग

पोर्ट पिन P1.0 आणि P1.1 हे L293D मोटर ड्रायव्हरच्या इनपुट पिन 2 आणि 7 ला जोडलेले आहेत. L293D, पिन 3 आणि 6 चे आउटपुट पिन DC मोटरला जोडलेले आहेत. आउटपुट टर्मिनल्स उलटे करून मोटरच्या रोटेशनची दिशा बदलता येते.

5.1.3 सी मध्ये डीसी मोटरचे प्रोग्रामिंग

अल्गोरिदम:

1. Initialize port P2 as output.
2. Send hex value 0x0C to port 2 and add delay
3. Send hex value 0x06 to port 2 and add delay.
4. Send hex value 0x03 to port 2 and add delay.
5. Send hex value 0x09 to port 2 and add delay.
6. Repeat from step2.

```
#include<reg51.h>
void delay(unsigned int t);
sbit motor_pin_1 = P1^0;
sbit motor_pin_2 = P1^1;
void main()
 {
while(1)
        {
   motor_pin_1 = 1;
   motor_pin_2 = 0; //मोटर उलट घड्याळाच्या दिशेने फिरते
delay(50);
   motor_pin_1 = 1;
   motor_pin_2 = 1; //मोटर थांबवते
delay(50);
   motor_pin_1 = 0;
   motor_pin_2 = 1; //मोटर घड्याळाच्या दिशेने फिरवते
delay(50);
   motor_pin_1 = 0;
   motor_pin_2 = 0; //मोटर थांबवते
delay(50);
 }
}
void delay(unsigned int t)
{
  inti, j;
  for (i=0; i<t; i++)
   for (j=0; j<1275; j++);
}
```

5.1.4 **सर्वो मोटर (Servo Motor) –**

सर्वो मोटर्स इलेक्ट्रॉनिक्स आणि एम्बेडेड सिस्टीममध्ये खूप उपयुक्त आहेत. तुमच्या आजूबाजूला सर्वत्र सर्वो मोटरचा वापर आढळू शकतो. सर्वो मोटर्स खेळणी, रोबोट, संगणकाच्या सीडी ट्रे, कार, विमान इत्यादींमध्ये वापरले जातात. या सर्वो मोटर्सच्यावापराचे कारण म्हणजे, सर्वो मोटर्स वापरायला खूप सोपेआणि अचूक आहे. आपण ते कोणत्याही विशिष्ट कोनात फिरवू शकतो.

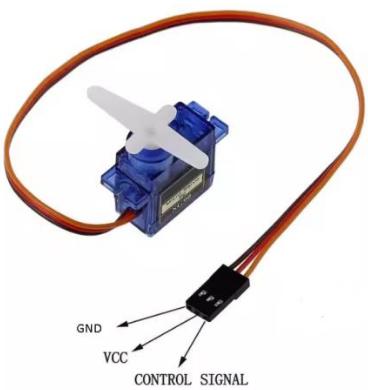


FIGURE 5.1.4 सर्वो मोटर

5.1.5 **सर्वो मोटरचे काम –**

सर्वो मोटर PWM (पल्स विड्थ मॉड्युलेशन) तत्त्वावर काम करते, म्हणजे त्याचा रोटेशन कोन त्याच्या कंट्रोल पिनवर लावलेल्या पल्सच्या कालावधीने नियंत्रित केला जातो. मुळात सर्वो मोटर DC मोटरपासून बनलेली असते जी व्हेरिएबल रेझिस्टर (पोटेंशियोमीटर) आणि काही गीअर्सद्वारे नियंत्रित केली जाते. DC मोटरचा हाय स्पीड फोर्स गीअर्सद्वारे टॉर्कमध्ये रूपांतरित केला जातो. आपल्याला माहित आहे की WORK = FORCE X DISTANCE, DC मोटरमध्ये फोर्स कमी आणि अंतर (वेग) जास्त असते आणि सर्वोमध्ये फोर्स जास्त असतो आणि अंतर कमी असते. कोन मोजण्यासाठी आणि आवश्यक कोनात DC मोटर थांबवण्यासाठी पोटेंशियोमीटर सर्वोच्या आउटपुट शाफ्टशी जोडलेला असतो. सर्वो मोटर 0 ते 180 अंशांपर्यंत फिरवता येते, परंतु मॅनुफॅक्चरींग रिंगवर अवलंबून ती 210 अंशांपर्यंत जाऊ शकते. 1ms ते 2ms दरम्यान कालावधीसाठी LOGIC लेव्हल 1 पल्स लावून रोटेशनची ही डिग्री नियंत्रित केली जाऊ शकते. 1 मिलिसेकंद सर्वोला 0 अंशापर्यंत, 1.5 मिलिसेकंद 90 अंशांपर्यंत आणि २ मिलिसेकंद पल्स 180 अंशांपर्यंत फिरवू शकते. 1 ते 2मिलिसेकंद कालावधी सर्वो मोटरला 0 ते 180 अंशांपर्यंत कोणत्याही कोनात फिरवू शकतो.

5.1.6 **8051 मायक्रोकंट्रोलरसह सर्वो मोटरचे इंटरफेसिंग –**

FIGURE 5.1.6 मध्ये सर्वो मोटरचे 8051 मायक्रोकंट्रोलरसह इंटरफेसिंग दाखवले आहे. सर्वो मोटरमध्ये तीन वायर आहेत ज्यात Vcc (पॉवर सप्लाय) साठी लाल, ग्राउंडसाठी तपकिरी आणि कंट्रोल वायर केशरी आहे. कंट्रोल वायर 8051 ला जोडता येते, आम्ही ती 8051 च्या पिन 2.1 शी जोडली आहे. आता आपल्याला हा पिन लॉजिक 1 ला 1 ms साठी 0 अंश फिरवण्यासाठी, 1.5 ms साठी 90 अंश, 2 ms साठी 180 अंश फिरवण्यासाठी ठेवावा लागेल.

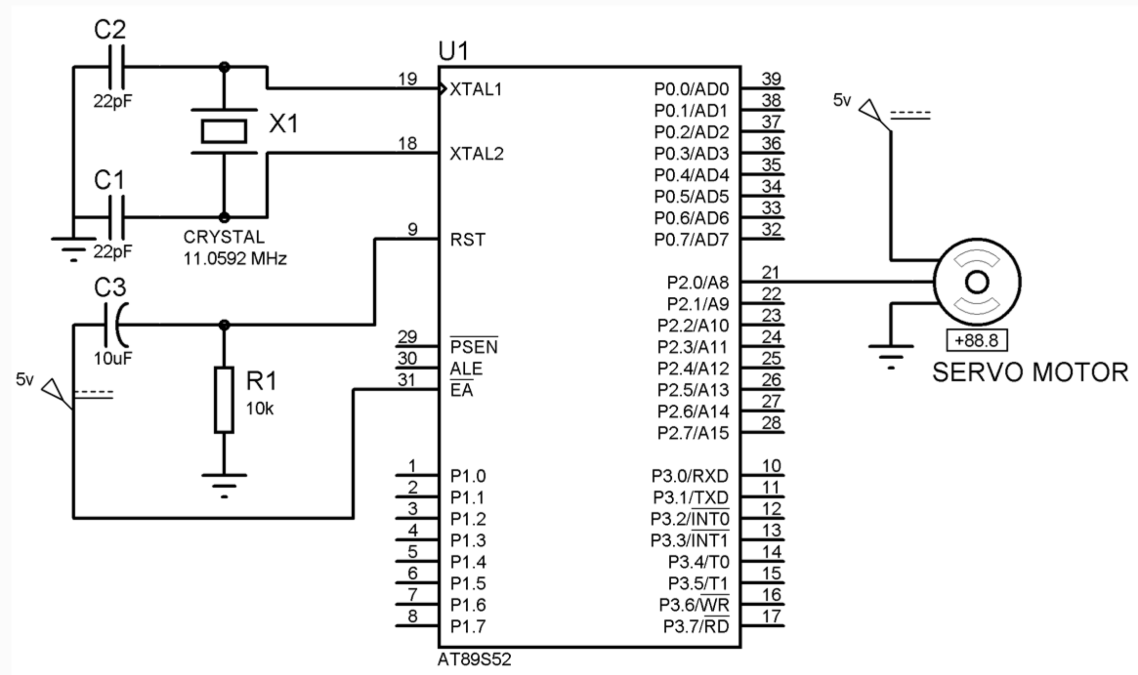


FIGURE.5.1.6 सर्वो मोटरचे 80C51 मायक्रोकंट्रोलरसह इंटरफेसिंग

5.1.7 सीमध्ये सर्वो मोटरचे प्रोग्रामिंग

अल्गोरिदम:

1. Set the servo control pin as output
2. ConFIGUREure Timer1 to generate PWM signals
3. Generate PWM Signal
   - Send 1ms pulse for 0°
   - Send 1.5ms pulse for 90°
   - Send 2ms pulse for 180°
4. Repeat the process continuously to control the servo motor.
5. Stop

```
#include<reg51.h>
sbit output=P2^0;
      void msdelay(unsigned int time)  // मिलिसेकंदांमध्ये विलंब निर्माण करण्यासाठी फंक्शन.
{
   unsigned i,j ;
   for(i=0;i<time;i++)
   for(j=0;j<1275;j++);
}
      void servo_delay(int times) // 8051टायमर वापरून 50us च्या मल्टीपलमध्ये विलंब निर्माण करणे
{
int m;
   for(m=0;m<times;m++)
   {
      TH0=0xFF;
      TL0=0xD2;
      TR0=1;
      while(TF0==0);
      TF0=0;
      TR0=0;
   }
}
void main()
{
int n;
TMOD=0x01;            // टाइमर 0 निवडणे, मोड 1
   output=0;
while(1)
   {
      for(n=13;n<28;n=n+2)
      {
         output=1;
servo_delay(n);
         output=0;
servo_delay(260);
msdelay(200);
      }
   }
}
```

5.2 स्टेपर मोटर (Stepper Motor) –

स्टेपर मोटर ही ब्रशलेस सिंक्रोनस मोटर असते जी पूर्ण रोटेशनला अनेक स्टेपमध्ये विभागते आणि प्रत्येक स्टेपर मोटरमध्ये काही निश्चित स्टेप अँगल असतो जो मोटर फिरवते. स्टेपर मोटर लहान कोनात फिरवली जाते, या कोनांना स्टेप्स म्हणतात. साधारणपणे स्टेपर मोटर 360 अंश रोटेशन पूर्ण करण्यासाठी 200स्टेप वापरते, म्हणजे ती प्रत्येक स्टेपवर 1.8 अंश फिरवते. आपण स्टेपर मोटरला योग्य सूचना देऊन कोणत्याही विशिष्ट कोनात फिरवू शकतो.

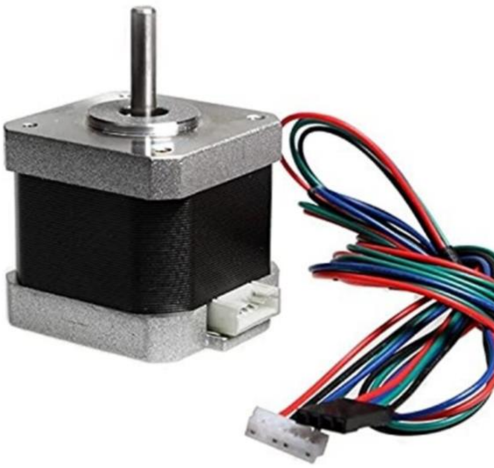


FIGURE 5.2 स्टेपर मोटर

रोबोट, अँटेना, ऑटोमॅटिक डोअर लॉक सिस्टीम, प्रिंटर, हार्ड ड्राइव्ह इत्यादीसारख्या अचूक रोटेशनल हालचालीची आवश्यकता असलेल्या अनेक उपकरणांमध्ये स्टेपर मोटर वापरली जाते. स्टेपर मोटर्स मुळात दोन प्रकारचे असतात: युनिपोलर आणि बायपोलर. युनिपोलर स्टेपर मोटरमध्ये साधारणपणे पाच किंवा सहा वायर असतात, ज्यामध्ये चार वायर चार स्टेटर कॉइल्सच्या एका टोकाला असतात आणि चारही कॉइल्सचे दुसरे टोक एकत्र बांधलेले असते जे पाचव्या वायरचे प्रतिनिधित्व करते, याला कॉमन वायर (कॉमन पॉइंट) म्हणतात. साधारणपणे दोन कॉमन वायर असतात, जे खालील FIGUREत दाखवल्याप्रमाणे दोन-दोन कॉइल्सच्या एका टोकाला जोडून तयार होतात. युनिपोलर स्टेपर मोटर वापरण्यास सोपी असल्याने खूप सामान्य आणि लोकप्रिय आहे. बायपोलर स्टेपर मोटरमध्ये कॉइलच्या दोन सेटमधून फक्त चार तारा बाहेर पडतात, म्हणजे कोणतेही सामान्य वायर नसते. स्टेपर मोटर स्टेटर आणि रोटेटरपासून बनलेली असते. स्टेटर म्हणजे चार इलेक्ट्रोमॅग्नेट कॉइल्स जे रोटेटरभोवती स्थिर राहतात आणि रोटेटर म्हणजे कायमचे चुंबक जे फिरते. जेव्हा जेव्हा कॉइल्स करंट लागू करून एंरजाईज्ड होतात तेव्हा इलेक्ट्रोमॅग्नेटिक फील्ड तयार होते, ज्यामुळे रोटेटर (कायमचे चुंबक) फिरतो. रोटेटर फिरवण्यासाठी कॉइल्स एका विशिष्ट क्रमाने एनर्जाइज केल्या पाहिजेत. स्टेपर मोटरमध्ये वापराच्या पद्धतीनुसार 0.9°, 1.8°, 2.0°, 2.5° इत्यादी वेगवेगळे स्टेप अँगल असतात.

स्टेपर मोटर्स खाली दिलेल्या दोन वेगवेगळ्या क्रमांमध्ये चालवता येतात:

1. हाफ स्टेपम्हणजेच0.9°
2. फुल स्टेप म्हणजेच 1.8°

फुल स्टेप मधे, एकाच वेळी दोन कॉइल्स ऊर्जावान होतात आणि मोटर शाफ्ट फिरते.

कॉइल्सना कोणत्या क्रमाने सिग्नल द्यावी लागते ते TABLE 5.2.1 मध्ये दिले आहे.

TABLE 5.2.1 फुल स्टेप क्रम

फुलस्टेपक्रम					**हेक्सव्हॅलू**
स्टेप	A	B	C	D	
0	1	1	0	0	0CH
1	0	1	1	0	06H
2	0	0	1	1	03H
3	1	0	0	1	09H

हाफ स्टेप मधे, मोटर स्टेप अँगल पूर्ण मोडमध्ये अर्ध्या कोनात म्हणजेच 0.9° पर्यंत कमी होतो, त्यामुळे कोनीय रिझोल्यूशन देखील वाढते, म्हणजेच फुल स्टेपच्या क्रमात कोनीय रिझोल्यूशनच्या दुप्पट. तसेच हाफ स्टेप मध्ये स्टेप संख्या फुल स्टेप मोडच्या तुलनेत दुप्पट होते आणि सामान्यतः पूर्ण मोडपेक्षा जास्त पसंत केली जाते. TABLE 5.2.2 कॉइल्सना ऊर्जा देण्याचे नमुने दर्शविते.

TABLE 5.2.2 हाफ स्टेप क्रम

हाफस्टेपक्रम					**हेक्सव्हॅलू**
स्टेप	A	B	C	D	
0	1	1	0	0	0CH
1	0	1	0	0	04H
2	0	1	1	0	06H
3	0	0	1	0	02H
4	0	0	1	1	03H
5	0	0	0	1	01H
6	1	0	0	1	09H
7	1	0	0	0	08H

5.2.1 स्टेपरमोटार ड्रायव्हर -स्टेपर मोटरला ULN2003 (एक सामान्य स्टेपर मोटर ड्रायव्हर IC) प्रमाणे ड्रायव्हर सर्किटची आवश्यकता असते. ULN2003 मायक्रोकंट्रोलरमधून स्टेपर मोटरच्या वायंडिंगमध्ये येणारा प्रवाह नियंत्रित करण्यास सक्षम आहे. FIGUREure 5.2.1मध्ये ULN2003 साठी पिन आउट FIGUREदाखवली आहे.

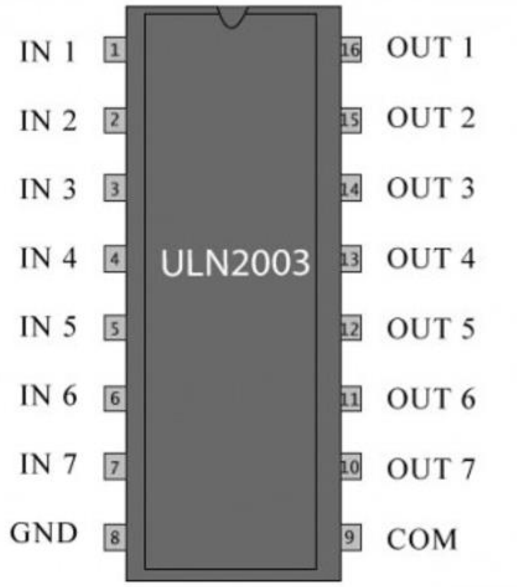


FIGURE 5.2.1 ULN2003 चा पिन

5.2.2 89C51 मायक्रोकंट्रोलरसह स्टेपर मोटरचे इंटरफेसिंग -FIGUREure 5.2.2 मध्ये ULN2003 डार्लिंग्टन अरे वापरून 8051 मायक्रोकंट्रोलरसह स्टेपर मोटरचे इंटरफेसिंग दाखवले आहे. पोर्ट 1 चे चार पोर्ट पिन P1.0, P1.1, P1.2 आणि P1.3 ULN2803 डार्लिंग्टन अॅरेशी जोडलेले आहेत.

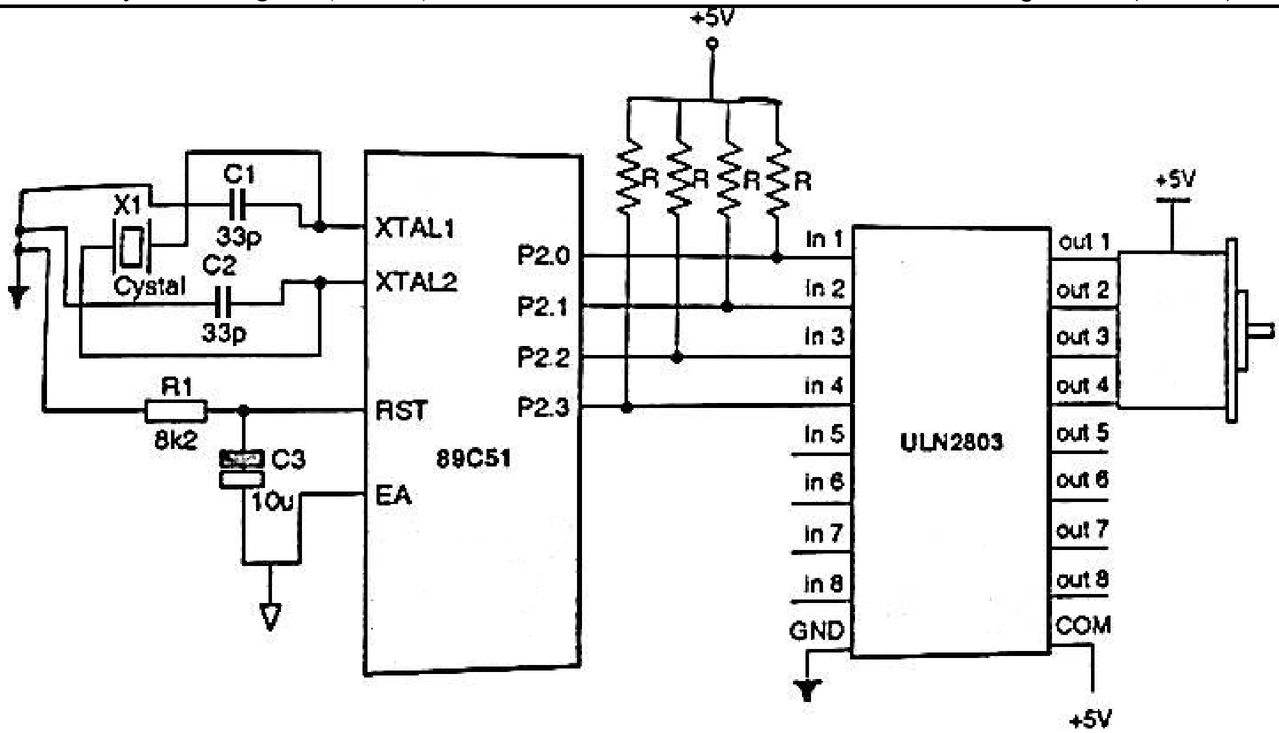


FIGURE. 5.2.2 ULN2003 वापरून स्टेपर मोटरचे 8051 मायक्रोकंट्रोलरसह इंटरफेसिंग

5.2.3 सी मध्ये स्टेपर मोटरचे प्रोग्रामिंग

अल्गोरिदम:

1. Initialize port P2 as output.
2. Send hex value 0x0C to port 2 and add delay
3. Send hex value 0x06 to port 2 and add delay.
4. Send hex value 0x03 to port 2 and add delay.
5. Send hex value 0x09 to port 2 and add delay.
6. Repeat from step2.

```
#include <reg51.h>
#define stepper P2
void delay(unsigned int t);
void main()
{
while(1)
        {
          stepper = 0x0C;
         delay(50);
          stepper = 0x06;
         delay(50);
         stepper = 0x03;
         delay(50);
         stepper = 0x09;
         delay(50);
        }
      }
}
```

```
void delay(unsigned int t)
{
unsigned char i, j;
for(i=0; i<t; i++)
for(j=0; j<1275; j++);
}
```

5.3 **आयआर सेन्सर (IR sensor)-** अवरक्त (Infrared) प्रकाश हा दृश्यमान लाल प्रकाशाच्या तुलनेत अधिक वेव्हलेन्थ (wavelength)असलेला प्रकाश आहे. अवरक्त श्रेणींमध्ये निअर-इन्फ्रारेड (Near-Infrared), मिड-इन्फ्रारेड (Mid-Infrared) आणि फार-इन्फ्रारेड (Far-Infrared) यांचा समावेश होतो. यामध्ये तरंगलांबी 710 नॅनोमीटर (Near-Infrared) पासून 100 मायक्रोमीटर (Far-Infrared) पर्यंत असते. सर्व वस्तू त्यांच्या तापमानानुसार प्रकाश उत्सर्जित करतात—यालाच "ब्लॅक बॉडी विकिरण" (Black Body Radiation) म्हणतात. वस्तू जितकी जास्त तापमानाची असेल, तितक्या कमी वेव्हलेन्थचा प्रकाश ती उत्सर्जित करते. पृथ्वी सुमारे 9 ते 10 मायक्रोमीटर वेव्हलेन्थचा अवरक्त प्रकाश उत्सर्जित करते आणि मानवी शरीरासारखी उष्ण-रक्तवाहिनी (Warm-Blooded) प्राणी देखील याच श्रेणीतील विकिरण उत्सर्जित करतात. हा प्रकाशहालचाली किंवा उष्णतेचा शोध घेण्यासाठी वापरला जाऊ शकतो. इन्फ्रारेड ऑब्स्टॅकल सेन्सर मॉड्यूलमध्ये बिल्ट-इन आयआर ट्रान्समीटर आणि आयआर रिसीव्हर आहे जो आयआर सिग्नल पाठवतो आणि सेन्सर मॉड्यूलसमोर कोणत्याही अडथळ्याची उपस्थिती शोधण्यासाठी परावर्तित IR ऊर्जा शोधतो.

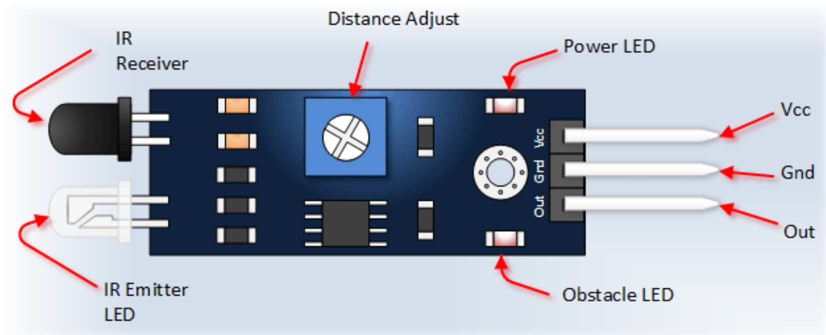


FIGURE 5.3 आयआर सेन्सर

5.3.1 **आयआर सेन्सरचे काम** -आयआर सेन्सरमध्ये एक आयआर एलईडी आणि एक आयआर फोटोडायोड असते; त्यांना एकत्रितपणे फोटो-कपलर किंवा ऑप्टो-कपलर असे म्हणतात. आधी सांगितल्याप्रमाणे, इन्फ्रारेड ऑब्स्टॅकल सेन्सरमध्ये एक बिल्ट-इन आयआर ट्रान्समीटर आणि आयआर रिसीव्हर असतो. इन्फ्रारेड ट्रान्समीटर हा एक प्रकाश-उत्सर्जक डायोड (एलईडी) असतो जो इन्फ्रारेड रेडिएशन उत्सर्जित करतो. म्हणूनच, त्यांना आयआर एलईडी म्हणतात. जरी आयआर एलईडी सामान्य एलईडीसारखा दिसत असला तरी, त्यातून उत्सर्जित होणारे रेडिएशन मानवी डोळ्यांना अदृश्य असते. FIGURE 5.3.1 आयआर सेन्सर वापरून अडथळा शोधणे दर्शविते.

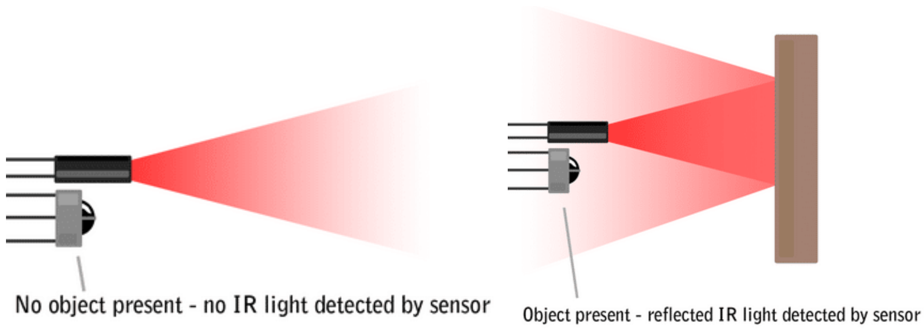


FIGURE 5.3.1 आयआर सेन्सर वापरून अडथळा शोधणे

इन्फ्रारेड रिसीव्हर्सना इन्फ्रारेड सेन्सर असेही म्हणतात कारण ते आयआर ट्रान्समीटरमधून रेडिएशन शोधतात. आयआर रिसीव्हर्स फोटोडायोड्स आणि फोटोट्रान्सिस्टर्सच्या स्वरूपात येतात. इन्फ्रारेड फोटोडायोड्स सामान्य फोटोडायोड्सपेक्षा वेगळे असतात कारण ते फक्त इन्फ्रारेड रेडिएशन शोधतात. जेव्हा आयआर ट्रान्समीटर रेडिएशन उत्सर्जित करतो तेव्हा ते

ऑब्जेक्टपर्यंत पोहोचते आणि काही रेडिएशन आयआर रिसीव्हरकडे परत परावर्तित होते. आयआर रिसीव्हरद्वारे रिसेप्शनच्या तीव्रतेवर आधारित, सेन्सरचे आउटपुट परिभाषित केले जाते.

5.3.2 8051 मायक्रोकंट्रोलरसह आयआर सेन्सरचेइंटरफेसिंग

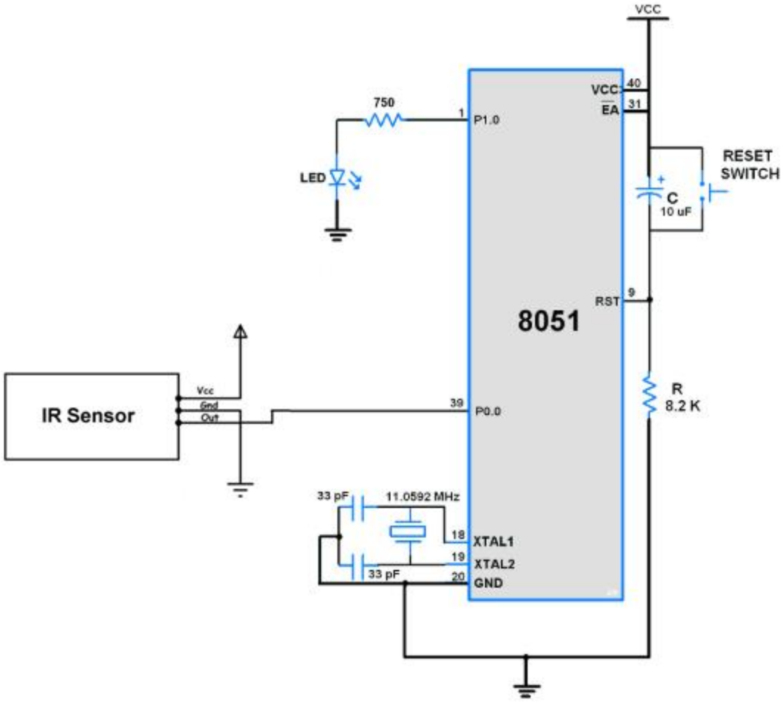


FIGURE 5.3.2 8051 मायक्रोकंट्रोलरसह आयआर सेन्सरचे इंटरफेसिंग

5.3.1 आयआर सेन्सर वापरून अडथळा शोधण्यासाठी प्रोग्राम

अल्गोरिदम:

1. Read IR sensor's data on this pin
2. Connect LED to the PORT1.0 pin
3. Initially LED turned OFF
4. Power-on delay for IR
5. Check for obstacle detection
6. LED turn OFF for No motion or LED turn ON if obstacle is detected
7. Repeat from step 5.

```
#include <reg51.h>
sbitobstacle_detection=P0^0;
sbit LED=P1^0;
void delay(unsigned int t);
void main(void)
{
    LED=0;
    delay(3000);
while(1)
    {
        if(obstacle_detection==1)
        LED = 0;
        else
        LED = 1;
```

```
        }
    }
    void delay(unsigned int t)
    {
        unsigned inti,j;
          for(i=0;i<=t;i++)
             for(j=0;j<1275;j++);        /* Delay of 1 ms for 11.0592MHz Frequency */
    }
```

5.4 पीआयआर सेन्सर (PIR sensor) -पीआयआर हा शब्द पॅसिव्ह इन्फ्रा-रेडचा संक्षिप्त रूप आहे. "पॅसिव्ह" हा शब्द सूचित करतो की सेन्सर प्रक्रियेत सक्रियपणे भाग घेत नाही, म्हणजेच तो संदर्भित आयआर सिग्नल स्वतः सिग्नल उत्सर्जित करत नाही, तर त्याऐवजी मानवी शरीरातून आसपासच्या परिसरात येणाऱ्या इन्फ्रारेड रेडिएशनचा शोध घेतो. शोधलेले रेडिएशन विद्युत चार्जमध्ये रूपांतरित केले जातात, जे रेडिएशनच्या शोधलेल्या पातळीच्या प्रमाणात असते. नंतर हे चार्ज बिल्ट-इन एफईटीद्वारे आणखी सुधारित केले जाते आणि डिव्हाइसच्या आउटपुट पिनला दिले जाते जे अलार्म स्टेजच्या पुढील ट्रिगरिंग आणि अँप्लिफिकेशन बाह्य सर्किटला लागू होते. पीआयआर सेन्सरची श्रेणी +150 किंवा -150 च्या कोनात 10 मीटर पर्यंत असते. बहुतेक पीआयआर मॉड्यूलमध्ये बाजूला किंवा तळाशी 3-पिन कनेक्शन असते.

FIGURE 5.4 पीआयआर सेन्सर

5.4.1 पीआयआर सेन्सरचे कार्य -पीआयआर सेन्सरमध्ये स्वतःच दोन स्लॉट असतात, प्रत्येक स्लॉट आयआरला संवेदनशील असलेल्या एका विशेष मटेरियलपासून बनलेला असतो. येथे वापरलेला लेन्स खरोखर फारसा काम करत नाही आणि म्हणून आपण पाहतो की दोन्ही स्लॉट काही अंतरावर (मुळात सेन्सरची संवेदनशीलता) 'पाहू' शकतात. जेव्हा सेन्सर निष्क्रिय असतो, तेव्हा दोन्ही स्लॉट्स एकसारख्या प्रमाणात अवरक्त (IR) किरणे शोधतात, जी खोली, भिंती किंवा बाहेरील वातावरणातून उत्सर्जित होतात. जेव्हा एखादे उष्ण शरीर, जसे की मनुष्य किंवा प्राणी, सेन्सरसमोरून जातो, तेव्हा प्रथम ते PIR सेन्सरच्या एका भागाला अडथळा आणते. यामुळे दोन्ही भागांमध्ये सकारात्मक भिन्नता निर्माण होते. जेव्हा ते उष्ण शरीर सेन्सिंग क्षेत्रातून निघून जाते, तेव्हा उलट प्रक्रिया घडते आणि सेन्सर नकारात्मक भिन्नता निर्माण करतो. याच बदलाच्या पल्सेस सेन्सरद्वारे शोधल्या जातात. FIGURE 5.4.1 पीआयआर सेन्सरची इंटर्नल रचना दर्शवते.

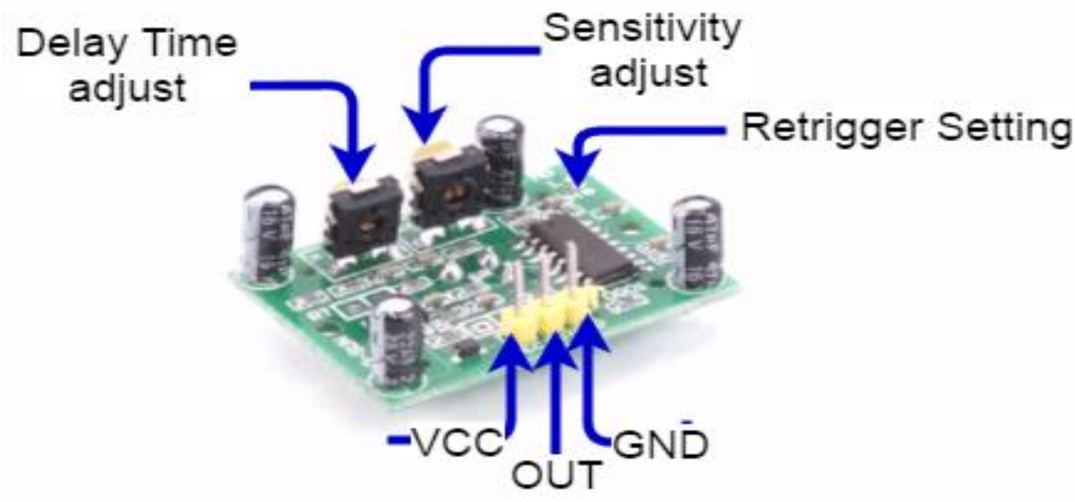


FIGURE 5.4.1 पीआयआर सेन्सरची अंतर्गत रचना

5.4.289C51 मायक्रोकंट्रोलरसह पीआयआर सेन्सरचे इंटरफेसिंग

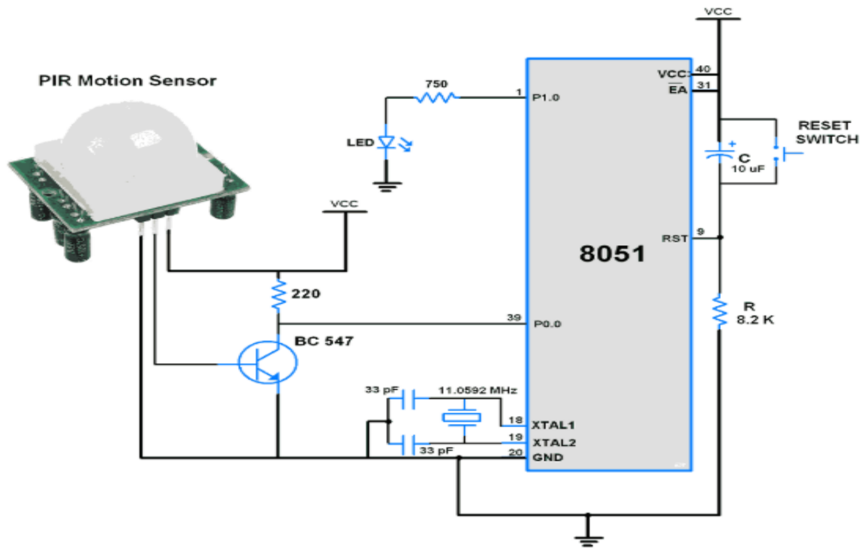


FIGURE. 5.4.2 मायक्रोकंट्रोलरसह पीआयआर सेन्सरचे इंटरफेसिंग

5.4.3 पीआयआर सेन्सर वापरून गती शोधण्यासाठी प्रोग्राम

अल्गोरिदम:

1. Read PIR sensor's data on this pin
2. Connect LED to the PORT1.0 pin
3. Initially LED turned OFF
4. Power-on delay for PIR
5. Check for human motion
6. LED turn OFF for No motion or LED turn ON if motion is detected
7. Repeat from step 5.

```
#include <reg51.h>
sbitMotion_detection=P0^0;        /* या पिनवर पीआयआर सेन्सरचा डेटा वाचा */
sbit LED=P1^0;                    /* LED ला PORT1.0 पिनशी जोडा */
void delay(unsigned int t);
void main(void)
{
```

```
        LED=0;                          /* सुरुवातीला एलईडी बंद होते*/
        delay(3000);    /* पीआयआरसाठी पॉवर-ऑन विलंब */
        while(1)
        {
                if(Motion_detection==1)  /* मानवी हालचाल तपासा */
                LED = 0;        /* हालचाल नसताना LED बंद करा */
                else
                LED = 1;        /* हालचाल आढळल्यास LED चालू करा */
        }
}
void delay(unsigned int t)
{
    unsigned inti,j;
      for(i=0;i<=t;i++)
         for(j=0;j<1275;j++);           /* 11.0592 MHz फ्रिक्केन्सीसाठी 1 मिलिसेकंद विलंब */
}
```

Exercise:

TLO 5.1: Develop ‘C’ program for rotating DC motor and servo motor interfaced with 8051.

1. Identify motor driver IC required for interface DC motor with 8051 microcontroller.(A)
2. Draw labeled diagram for interfacing DC motor with 8051 microcontroller.(A)
3. Write C language program to rotate DC motor in clockwise direction with neat interfacing of DC motor with 8051 microcontroller.(A)
4. Write ‘C’ program for rotating DC motor in counterclockwise direction with neat interfacing diagram.(A)
5. Draw labeled interfacing diagram to interface servo motor with 8051 microcontroller.(A)
6. Develop ‘C’ program to rotate servo motor by 90° clockwise direction interfaced with 8051.(A)

TLO 5.2: Develop ‘C’ program for rotating stepper motor in clockwise and anticlockwise direction.

1. Identify motor driver IC required to interface stepper motor with 8051 microcontroller.(A)
2. Draw the interfacing diagram of stepper motor with 8051 microcontroller.(A)
3. Write ‘C’ language program to rotate the stepper motor in clockwise direction.(A)
4. Sketch interfacing diagram to control stepper motor connected to port 2 through IC ULN 2003 and write C language program to rotate stepper motor in clockwise direction continuously with certain delay(A)
5. Write a ‘C’ language program to rotate stepper motor by 90° in clockwise direction. Assume step angle of 1.8° and 4 step sequences. (A)
6. Write ‘C’ language program to rotate stepper motor in anticlockwise direction interfaced with 8051 microcontroller.(A)

TLO 5.3: Develop ‘C’ program for obstacle detection using IR sensor.

1. Draw the interfacing diagram of IR sensor with 8051 microcontroller.(A)
2. Write C language program to read IR sensor value and put it on port P1 of 8051 microcontroller, also draw the interfacing diagram of it.(A)

TLO 5.4: Develop ‘C’ program for motion detection using PIR sensor.

1. Draw the interfacing diagram of PIR sensor with microcontroller 8051.(A)
2. Write C language program to read PIR sensor value and put it on port P1 of 8051 microcontroller, also draw the interfacing diagram of it.(A)

Bibliography:

1. **SUGGESTED LEARNING MATERIALS / BOOKS**

Sr.No	Author	Title	Publisher with ISBN Number
1	Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin D. McKinlay	The 8051 Microcontroller and Embedded Systems using Assembly and 'C'	Pearson, 2007 ISBN-13: 978-0199681273
2	Kenneth J. Ayala	The 8051 Microcontroller: Architecture, Programming and Applications	Penram International Publishing, 1996ISBN-13: 978-0314201881
3	Raj Kamal	Embedded Systems	McGraw Hill, 4th Edition, 2020ISBN-13: 978-9353168025
4	Ajit Pal	Microcontrollers: Principles and Applications	PHI Learning Pvt. Ltd., 2011ISBN-13: 978-8120343924
5	Ajay V. Deshmukh	Microcontrollers: Theory and Applications	McGraw-Hill Education (India) Pvt. Ltd., 2005ISBN-13: 978-0070585959

1. **LEARNING WEBSITES & PORTALS**

Sr.No	Link / Portal	Description
1	https://www.keil.com/demo/eval/c51.htm	Keil IDE download
2	https://www.engineersgarage.com/timers-8051-timer-programming/	8051 timer programming
3	https://www.tutorialspoint.com/embedded_systems/es_io_programming.htm	8051 I/O programming
4	https://electrosome.com/interfacing-relay-8051-keil-c/	Relay interfacing and programming
5	https://www.javatpoint.com/embedded-system-7-segment-display	7-segment-display interfacing and programming
6	https://www.javatpoint.com/embedded-system-lcd-programming	LCD interfacing and programming
7	https://www.elprocus.com/embedded-system-programming-using-keil-c-language/	Embedded 'C' programming
8	https://www.electronicwings.com/8051/dc-motor-interfacing-with-8051	DC-motor-interfacing-with-8051
9	https://electrosome.com/interfacing-stepper-motor-8051-keil-c-at89c51/	Stepper motor interfacing and programming
10	https://circuitdigest.com/microcontroller-projects/servo-motor-interfacing-with-8051	Servo motor interfacing and programming

11	https://embtronix.com/tutorials/microcontrollers/8051/ir-sensor-interfacing-with-8051/	IR sensor interfacing and programming
12	https://www.electronicwings.com/8051/pir-motion-sensor-interface-with-8051	PIR sensor interfacing and programming
13	https://vlabs.iitkgp.ac.in/rtes/index.html	Virtual Lab. for embedded system