INPLANT TRAINING REPORT

Diploma in Electronics Semester VIII

Submitted for the Partial fulfillment of the course ‘Diploma in Electronics’ from
Shreemati Nathibai Damodar Thackersey Women’s University Mumbai

Miss. Kapadnis Sakshi. (GR No.17070104)

Miss. Vekariya Manasvi. (GR No.17070114) At

Tata Institute of Fundamental Research

1, Dr. Homi Bhabha Road, Navy Nagar, Colaba, Mumbai 400 005.

(Form Monday; 6 May, 2019 to Saturday; 19 October, 19, 2019 – (24 weeks))

Department of Electronics

PREMILA VITHALDAS POLYTECHNIC

Academic Year 2019 – 20

S. N. D. T. Women’s University

Sir Vithaldas Vidyavihar, Juhu Road, Santacruz (West) Mumbai 400 049.

ACKNOWLEDGEMENT

It is indeed a moment of great pleasure and immense satisfaction for us to express our sense of profound gratitude and indebtedness to all the people who have contributed in making out training a rich and meaningful experience.

“EXPERIENCE IS THE BEST TEACHER.”

We are grateful to our parent institute P.V. POLYTECHNIC for giving us this opportunity to undergo six month long In-plant training at The LinAc Facility of Tata Institute of Fundamental Research (TIFR).

Nothing crystallizes in our mind except the enthusiasm and personal interest taken by our guide Shri. Jitendra. N. Karande, S. O. – ‘D’. (at TIFR) and his colleagues. Their constant motivation, timely guidance, critical suggestion and co-operation helped us in overcoming various theoretical as well as practical problems which we encountered during the tenure of our training.

Above all, we express our deepest gratitude to all the staff members of the Linac Facility for their kind- hearted support, which helped us during our training. They offered us plenty of opportunities while working with them, rendered us invaluable help and helped linking practical knowledge with theory which was taught to us in the college.

We also express our gratefulness to our Head of the Department Shri. Dinesh Girap and to our training supervisor Shri. Sanjay Kumar H. Desai. Without their constant encouragement, our training would not have led to conclusion.

Finally, we would like acknowledge the help extended by the entire staff of Pelletron-LinAc Facility at TIFR with whom we worked during our training.

We shall always cherish very fondly the learning experience we have had at TIFR.

Table of Contents **Page no**

Table of Contents **Page no**

Table of Contents **Page no**

1. Objectives of In-plant Training

The classroom teaching alone is not sufficient to prepare effective technicians; it has to go along with the effective institute – industry relation. Thus, it is imperative that the polytechnic should extend its activities beyond the limit of classroom teaching and make every effort to include industrial work culture in the student. The aspiring technicians are exposed to the industry through in – plant training. The trainee should go to the industry with the following objectives,

- To understand the work culture of the industry.
- To understand the changing needs of the society.
- To relate their knowledge with its industrial applications.
- To improve the basic skills of the technicians like designing, testing, tracing, debugging, soldering, documenting and many more.
- To tackle the challenges successfully.
- To develop entrepreneurial spirit among trainee.

The In–plant Training of semester V of diploma in electronics engineering helps the student to relate two years knowledge with the industrial applications. The training helps the students to know how the device works and also how a particular component learned in the institute works like transistor, diode and many more active and passive devices. The knowledge gained helps students in making their own working projects which requires many steps including designing, testing, soldering and many more. Thus, the whole purpose of In-plant Training is to develop the thinking which in turn can fulfill industrial requirement for the job as an employee.

2. Introduction to Industry

Tata Institute of Fundamental Research:

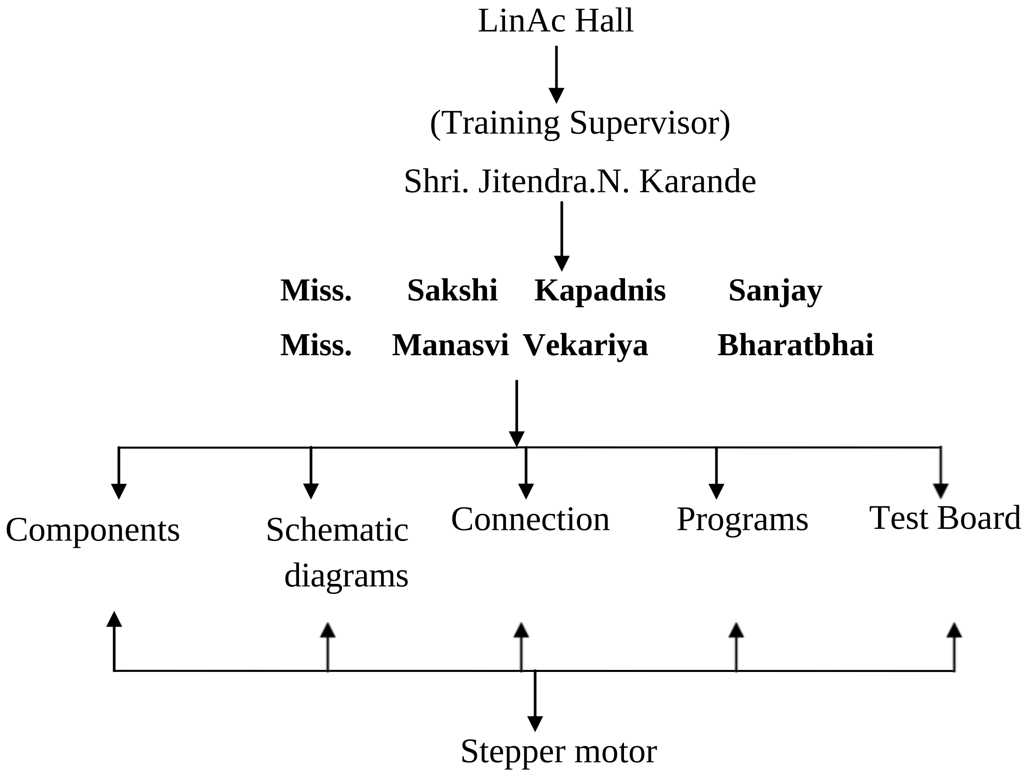
The **Tata Institute of Fundamental Research (TIFR)** is a premier institution in India for higher education and research. The main academic disciplines studied at the institute are natural sciences, mathematics and theoretical computer science. It is located at Navy Nagar, Colaba, Mumbai.

It is established in 1945 under the direction of Homi J. Bhabha with the help of J. R. D. Tata. It is funded primarily by the Government of India. For most of its history, it had been affiliated to the University of Mumbai but like all other premier research institutes in country it has always enjoyed academic and administrative autonomy. It attained the official deemed university status in June 2002. There are at present about 400 scientists in the Institute working in various disciplines.

It is one of the outstanding research centers in India, and is regarded as being a leader in certain fields like Mathematics and String Theory. TIFR has a graduate program leading to a PhD in all the major fields of study. The graduate program is selective and the selection is based on a written exam followed by interviews.

On the technological side, TIFR pioneered activity in the field of computer technology by designing India's first computer (TIFRAC). Group working in other emerging fields such as accelerators, microwave communications, software technology, semiconductor technology and educational research were also set up at the institute.

3. Work Environment of the Industry



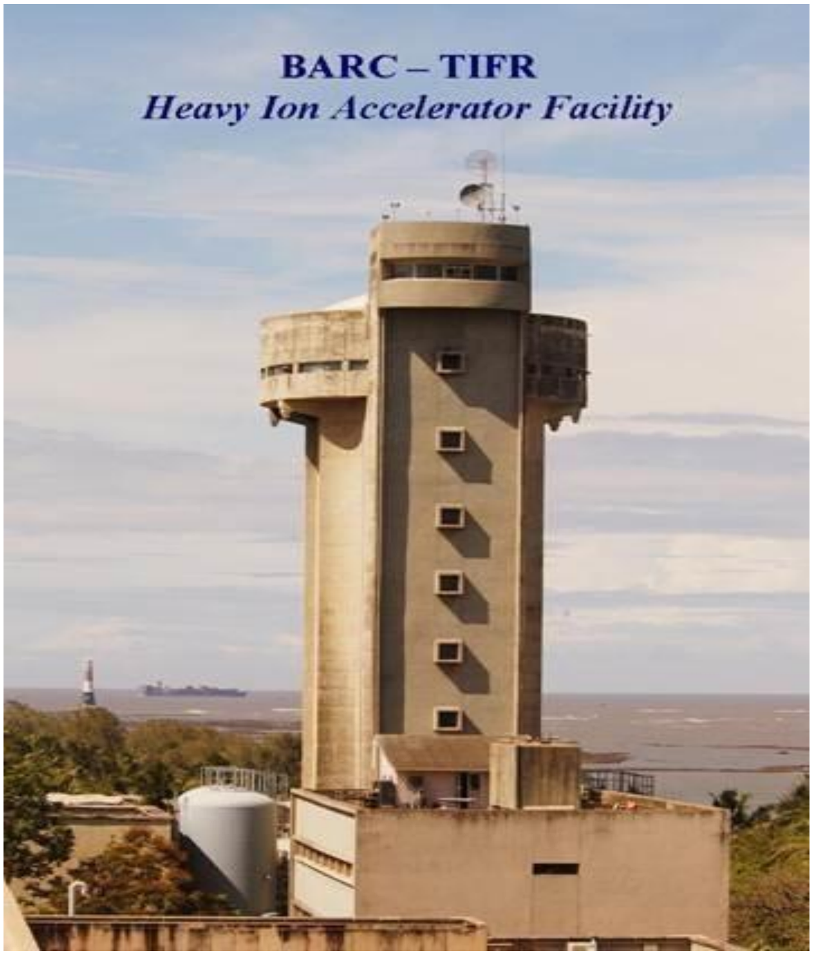
➢ Introduction of Particle Accelerator

A particle accelerator is a machine that uses electromagnetic fields to propel charged particles to move nearly at the speed of light and to contain them in well define beams. There are two basic classes of linear accelerators**:** (a) Electrostatic Accelerator and (b) Electrodynamic (or electromagnetic) Accelerator. An Electrostatic accelerator uses static electric fields to accelerate charged particles. The most common types are Cockcroft-Walton generator and the Van-de- Graff generator. A small-scale example of this class is the cathode ray tube in an ordinary old time television receiver.

- Principle of Particle Accelerator:

Particle accelerators exist in many shapes and sizes, but the smallest accelerator shares common elements with the large devices. All accelerators must have electric field to accelerate charged particles and they must have magnetic fields to control the path of the particles. Also, the particles must travel through a good vacuum i.e. in a container with as little residual air as possible as in a television tube. Finally, all accelerators must have some means of detecting; counting and measuring the particles after they have been accelerated through the vacuum

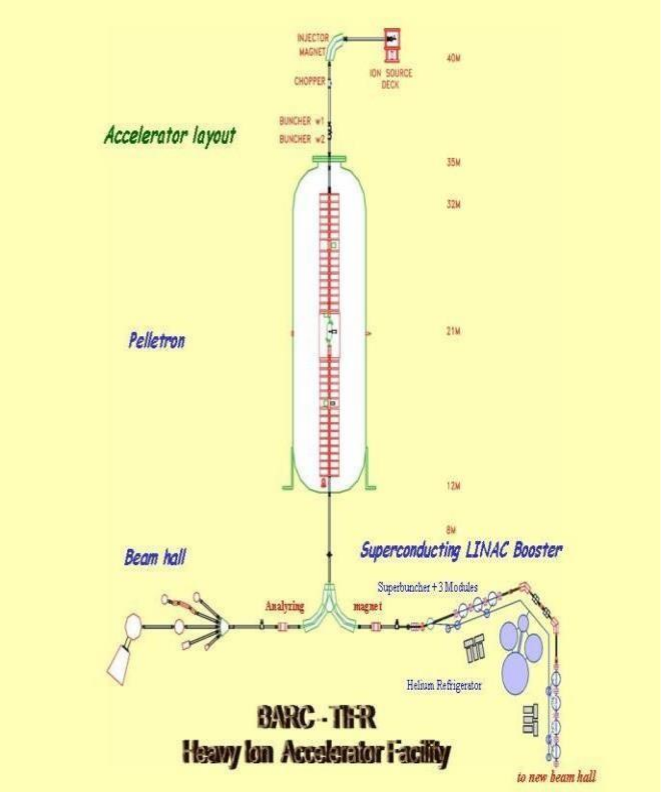
Linac Booster for the Pelletron Accelerator



The pelletron accelerator, setup as a collaborative project between the Bhabha Atomic Research Centre (BARC) and Tata Institute of Fundamental Research (TIFR); it has been serving as a major facility for heavy ion accelerator-based research in India since its commissioning in 1989. Several advanced experimental facilities have also been established at this centre to pursue research in Nuclear, Atomic, Condensed matter and Bio-Environmental Physics. While majority of researchers at this facility are from BARC and TIFR, the experimental community encompasses scientists and student from other research centers and universities within and outside the country. The research work in Nuclear Physics which forms the main thrust of activities at this facility covers area of nuclear structure studies at high temperature and angular momentum, elastic and non-elastic energy transfer reactions as well as fusion-fission reactions.

These past years have been scientifically stimulating and very productive. More than 50 Ph. D. thesis and over 350 publications in referred international journals including 12 publication in physical review letters have resulted from the research activities in above the laboratory.

Superconducting Linac Booster for the Pelletron Accelerator at TIFR



A super-conducting linear accelerator has been indigenously developed to boost the energy of the heavy ion beams delivered by the pelletron accelerator. The Linac Booster consists of seven modules. Each module is a liquid helium cryostat housing four lead coated copper quarter wave resonators. The superconducting LinAc Booster Phase-I consisting of three accelerating modules was commissioned on September 22, 2002, when silicon beam of 85 MeV from the pelletron was accelerated through linac to 130 MeV. Subsequently, the linac was smoothly operated for a period of about three months for several beam diagnostic test and for the first user cycles of two experiments. The work on phase-2 four modules is nearing completion. In May 2006, silicon beam was accelerated using two modules of the second phase and the beam was transported to one of the experimental stations in the new experimental hall. These initial beam trials have given excellent results with an average energy gain ~90% of design value. Most of the critical components of linac booster, the first superconducting heavy ion accelerator in India, have been designed and developed indigenously.

4. Work done by us

4.1 Study of Microprocessor and Microcontroller

Microprocessor and microcontroller stem from the same basic idea, are made by the same people, and are sold to the same type of system designers and programmers.

4.1.1 Microprocessor

A microprocessor is a general-purpose digital computer central processing unit (CPU). Although popularly known as a "Computer on a chip," the microprocessor is in no sense a complete digital computer.

Fig 01 shows the block diagram of a general microprocessor-based system which contains a general purpose microprocessor (that is, CPU), RAM, ROM, I/O Port/s, Timer, Serial COM Port and Address as well as Data bus.

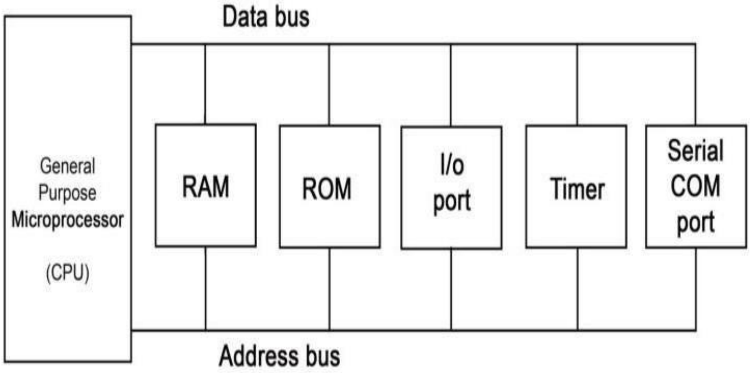


Figure 01: Block diagram of Generic Microprocessor

The prime use of microprocessor is to read data, perform extensive calculations on that data, and to store those calculations in a mass storage device or display the results for use. The programs used by the microprocessor is stored in the mass storage device and loaded into the RAM as per the user directives.

To make a complete microcomputer based system, one must add memory, usually read-only program memory (ROM) and random-access data memory (RAM), memory decoders, an oscillator, and a number of input/output (I/O) devices, such as parallel and serial data ports. In addition, special-purpose devices, such as interrupt handlers and counters, may be added to relieve the CPU from time-consuming counting or timing routines.

4.1.2 Microcontroller:

A microcontroller is a small computer on a single integrated circuit. Fig.02 shows the block diagram of a typical microcontroller, which is a true computer on a chip. The design incorporates all of the features found in a microprocessor CPU: ALU, PC, SP, and registers. It also has added the other features needed to make a complete computer: ROM, RAM, parallel I/O, serial I/O, counters and a clock circuit. Like a microprocessor, a microcontroller is a general-purpose device, but the one which is meant to read data perform limited calculations on that data, and to control its environment based on those calculations.

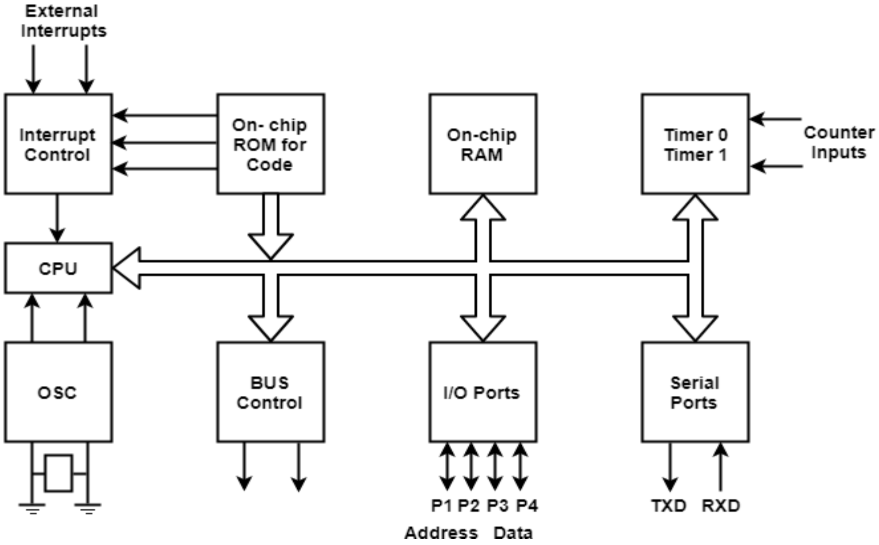


Figure 02: Block diagram of microcontroller 8051

A microcontroller is a true computer on a chip. The prime use of a microcontroller is to control the operation of a machine using a fixed program that is stored in its ROM which does not change over the lifetime of the system. The microcontroller design uses a much more limited set of single and double byte instructions that are used to move code and data from the internal memory to the ALU. Many instructions are coupled with pins on the integrated circuit package; the pins are “programmable” that is, capable of having several different functions depending on the wishes of the programmer.

Comparing Microprocessor and Microcontroller:

Microprocessor	Microcontroller
Microprocessor is a single chip CPU of microcomputer.	Microcontroller is a single chip microcomputer.
Microprocessor contains ALU, General Purpose Registers, Stack Pointer, Program Counter, Clock Timing Circuit, and Interrupt Circuit.	Microcontroller contains the circuitry of microprocessor, in addition it has in-built RAM, ROM, I/O devices, Timers/Counter.
It has few bits handling instruction.	It has many bits handling system.
Less number of pins is multi-functional.	More number of pins is multifunctional.

It is used in general purpose computer system.	It is used in specific system.
More flexible to design using microprocessor.	Less flexible to design using microcontroller.
Access time for external memory and I/O devices are more, resulting in a slower system.	Access time for on-chip memory and I/O devices are less, resulting in a faster system.
System require support devices and are usually bulky, costly, less reliable and consume more power.	System require less external hardware, reducing PCB size, and hence are compact, cheaper, more reliable and consume less power.
Software protection is not possible because of the requirement of external code memory.	Software protection is possible because of the on-chip memory.
Microprocessor is based on Von-Neumann architecture.	Microcontroller is based on Harvard architecture.

Types of Microcontroller: They are characterized by their bits, memory architecture, memory/devices and instruction set.

MICROCONTROLLERS

8051 (year 1984)	**AVR** (year 1996)	**PIC** (year 1976)	**ARM** (year 1985)
89S52	ATMEGA16	PIC10FXX	ARMv8-A
89S51	ATMEGA08	PIC12FXX	ARMv8.1-A
89C52	ATMEGA32	PIC16FXX	ARMv8.2-A
89C51	list continues…	list continues…	ARMv8.3-A
89V51XX			
List continues**…**			

4.1.2.1 Classification of Microcontroller 8051 According to Number of Bits

The number bits in a microcontroller are 8-bits, 16-bits and 32-bits.

An **8-bit** microcontroller consists of internal bus which is 8-bits wide and the ALU is performs the arithmetic and logic operations. The examples of 8-bit microcontrollers are Intel 8031/8051, PIC16xx and Motorola MC68HC11 families.

A **16-bit** microcontroller performs operations with greater precision and performance as compared to 8-bit. For example, 8-bit microcontrollers can only use 8-bits, resulting in a final range of 0x00-0xFF (0-255) for every cycle. In contrast, 16-bit microcontrollers with its 16-bit data width have a range of 0x0000-0xFFFF (0- 65535) every cycle. A longer timer most extreme worth can likely prove to be useful in certain applications and circuits. It can automatically operate on two 16-bit numbers. Some examples of 16-bit microcontroller are 16-bit MCUs are extended 8051XA, PIC2x, Intel 8096 and Motorola MC68HC12 families.

A **32-bit** microcontroller uses the 32-bit word to perform the arithmetic and logical operations. These are used in automatically controlled devices including implantable medical devices, engine control system, office machine, applications and other types of embedded systems. Some examples are Intel MCS-251 family, PIC3X.

4.1.2.2 Classification of Microcontroller 8051 According to memory devices

These memory devices are divided into two types; they are**:**

- Embedded memory microcontrollers and
- External memory microcontrollers

Embedded memory microcontrollers: When an embedded system has a microcontroller unit that has all the functional blocks available on a chip is called an embedded microcontroller. For example, 8051 having program & data memory, I/O ports, serial communication, counters and timers and interrupts on the chip is an embedded microcontroller.

External memory microcontroller: When an embedded system has a microcontroller unit that has not all the functional blocks available on a chip is calls an external memory microcontroller. For example, 8031 has no program memory on the chip is an external memory microcontroller.

4.1.2.3 Classification of Microcontroller According to the Size of Instruction Set

CISC Processor: CISC is a Complex Instruction set Computer. It allows the programmer to use one instruction in place of many simpler instructions.

RISC Processor: The RISC is stands for Reduced Instruction set Computer; this type of instruction sets reduce the design of microprocessor for industry standards. It allows each instruction to operate on any register or use any addressing mode and simultaneous access of program and data.

8051 has a simple instruction set in different groups. They are:

- Arithmetic instructions
- Logical instructions
- Data Transfer instructions
- Branching and looping instructions
- Boolean Variable instructions

➢ **Arithmetic Instruction:** These instructions are used to perform various mathematical operations like addition, subtraction, multiplication and division etc.

➢ **Logical Instructions:** The logical instructions are the instructions which are used for performing some operations like AND, OR, NOT, X–OR and etc., on operands.

➢ **Data Transfer Instruction:** These instructions are used to transfer the data from source operand to destination operand. All the store, move, load, exchange inputs and output instruction belong to this group.

➢ **Branching and Looping Instruction:** These instructions are used for both branching and looping. These instructions include conditional and unconditional jump or loop instructions.

➢ **Boolean Variable Instruction:** These instructions allow manipulating the individual bits with bit addressable registers and memory locations as well as the CY flag.

4.1.2.4 Classification of Microcontroller According to Memory Architecture

Memory architecture of microcontrollers are of two types, they are namely:

- Princeton memory (Von-Neuman) architecture microcontrollers and
- Harvard memory architecture microcontrollers

Comparing of Von-Neumann and Harvard Architecture Microcontrollers

Table 02: Comparison between Von-Neumann and Harvard Architecture Microcontrollers

Von Neumann (Microprocessor)	Harvard (Microcontroller)
Single address and data bus are used to connect program memory (ROM) and data memory (RAM).	Separate buses are used for connecting program memory (ROM) and data memory (RAM).
Accessing from data memory and program memory cannot be done in parallel.	Accessing from data memory and program memory can be done in parallel.
Speed of execution is slow.	Speed of execution is fast.
Use CISC architecture.	Use RISC architecture.

4.1.2.5 Features of Microcontroller 8051

- On chip clock oscillator
- 4K bytes of internal program memory (ROM)
- 128 bytes of internal data memory (RAM)
- 64Kbytes of external data memory address space
- 64Kbytes of external program memory address space
- 32 bi-directional I/O lines (can be used as four I/O ports)
- Two 16-bit Timer/Counter: T0, T1
- Full duplex serial data receiver/transmitter
- Four register banks with 8 registers in each bank
- 16-bit Program Counter (PC) and a Data Pointer (DPTR)
- 8-bit Program Status Word (PSW)
- 8-bit Stack Pointer (SP)

4.1.2.6 Pin Diagram and Description of Intel 8051:

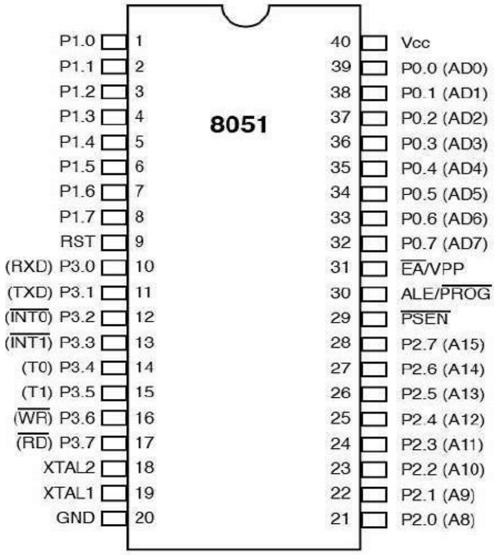


Figure 03: Pin diagram of 8051

1. **Pins 01 to 08(Port 1):** Pins from 01 to 08 are known as Port 1. Port 1 pins consist of 8-bits bidirectional input/output pins with internal pull-up resistors.
2. **Pin 09:** Pin 09 is known as Reset pin, which is used to reset the microcontroller to its initial values.
3. **Pin 10 to 17(Port 3):** Pins from 10 to 17 are known as Port 3. Port 3 pins consist of 8-bits

bidirectional input/output pins with internal pull-up resistors. Port 3 pins serve additional functions. The following table gives the details of the additional functions of the Port 3 pins:

Table 03: Alternative Functions of Port 3

Port 3 Pins	***Functions***	***Description***
P3.0	RXD	Serial input
P3.1	TXD	Serial output
P3.2	_____	External Interrupt 0
P3.3	_____	External Interrupt 1
P3.4	T0	Timer 0
P3.5	T1	Timer 1
P3.6	_____	External Memory Write
P3.7	_____	External Memory Read

4. **Pins 18 & 19:** Pins 18 & 19 are known as external oscillator XTAL0 and XTAL1 respectively.
5. **Pin 20:** Pin 20 is Ground pin for 8051 and connected to the negative part of the power supply.
6. **Pin 21 to 28(Port 2):** Pins from 21 to 28 are known as Port 2. Port 2 pins consist of 8-bits bidirectional input/output pins with internal pull-up resistors. Additionally, when external memory is interfaced Port 2 Pins are act as higher order address byte.
7. **Pin 29:** Pin 29 is known as Program Store Enable pin (). By using this pin external memory can be read.
8. **Pin 30:** Pin 30 is known as Address Latch Enable pin (ALE). By using this pin external address can be separated from data.
9. **Pin 31:** Pin 31 is known as Enable pin). Allow external program memory.
10. **Pin 32 to 39(Port 0):** Pins from 32 to 39 are known as Port 0. Port 0 pins consist of 8-bits bidirectional input/output pins but without internal pull-up resistors. Hence, we need external pull-up resistors in order to use Port 0 pins as I/O port.
11. **Pin 40:** Pin 40 is the power supply pin.

4.1.2.7 Block diagram and Architecture of Microcontroller 8051:

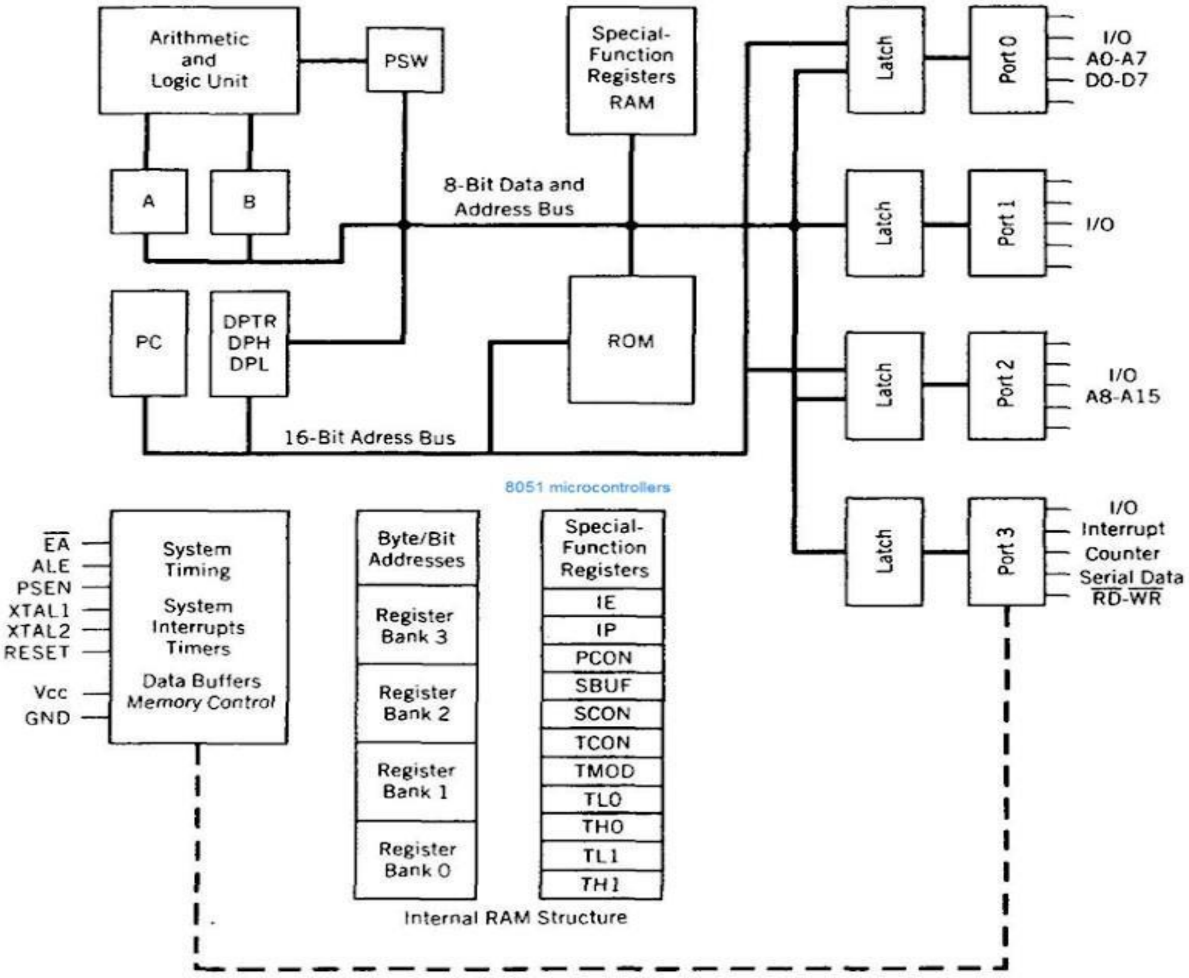


Figure 04: Block diagram and Architecture of Microcontroller 8051

The Architecture of 8051 can be understood from the block diagram. The block diagram of 8051 microcontroller is shown in fig .03. It consists of 8-bit ALU, A and B resistor, one 8-bit Program status word (PSW), one 16-bit Program counter (PC), one 16-bit Data pointer register (DPTR), 128-bytes of RAM and 5KB of ROM, four parallel I/O ports each 8-bit width.

4.1.2.7.1 Description of Registers of 8051

➢ **Arithmetic and logic unit (ALU):**

- It is an 8-bit register. It performs an arithmetic operation such as addition, subtraction, multiplication, division, increment and decrement.
- It performs logical operations such as AND, OR, and EX-OR.
- It manipulates 8-bit and 16-bit data.
- ALU contains 32 general purpose register or working register. 2 of them are called math registers A and B and 32 are bank of register.

- **A and B registers:**
  - A and B registers are each 8-bit registers.
  - Addresses of register A and B are E0H and F0H respectively.
  - Both registers are bit and byte addressable.
  - Register A is used to store 8-bit data and to hold one of operand of ALU unit during arithmetic and logic operations.
  - Register B is used as general-purpose register to store 8-bit data.
- **R registers:**
  - The R registers are a set of eight registers that are named from R0 to R7.
  - These registers are used as auxiliary register in many operations.
  - The R register are also used to temporarily store value.
- **Program counter (PC):**
  - Program counter is a 16-bit register.
  - The program counter points to the address of the next instruction to be executed.
  - After execution of one address the program counter is incremented to point the address of the next instruction to be executed.
  - Since, the PC is 16-bit width, 8051 can access program addresses from 0000H to FFFFH.
- **Data pointer register (DPTR):**
  - DPTR is a 16-bit register used to hold address of internal and external RAM.
  - It is used to store 16-bit data.
  - It is divided into two 8-bit registers i.e. data pointer high (DPH) and data pointer low (DPL).
  - Addresses of DPH and DPL are 83 H and 82H respectively. DPTR does not have single internal address.
  - Each register can be used as general-purpose register to store 8-bit data and can also be used as memory location.
- **Stack pointer (SP):**
  - Stack pointer is an 8-bit register. It is byte addressable. It addresses is 81H.
  - It is used to hold the internal RAM memory location addresses which are used as stack memory.
  - When the data is to be placed in stack by push instruction, the content of stack pointer is incremented by 1.

- When data is retrieved from stack, contented of stack, of stack pointer is decremented by 1.

➢ Program status word (PSW):

- PSW is 8-bit register also known as flag register.
- Only 6-bits of PSW are used by 8051 and the 2 unused bits are user definable bits.
- In 6-bits 4 of them are conditional flags. They are Carry (CY), Auxiliary Carry (AC), Parity (P) and Over Flow (OV).
- Flag bits indicate some condition that resulted after an instruction was executed.

PSW.7	PSW.6	PSW.5	PSW.4	PSW.3	PSW.2	PSW.1	PSW.0
CY	AC	F0	RS1	RS0	OV	-	P

Table No 04: Bit-pattern of Program Status word

I. PSW.7(CY): Carry flag

- This flag is set whenever there is carry out from the D7 bit.
- This flag bit is affected after 8-bit addition and subtraction.

II. PSW.6(AC): Auxiliary flag

- If there is a carry from D3 to D4 during an ADD and SUB operation, this bit is set, otherwise, it is reset.
- This flag is used by instructions that perform BCD (binary coded decimal) arithmetic.

III. PSW.5(F0):user flag

- It is user defined flag. The user defines the function of this flag.
- The user can set, test and clear this flag through software.

IV. PSW.4 and PSW.3 (RS1 and RS0): Register bank select bit 1 & 0.

RS1	RS0	Register Bank	Address
0	0	0	00H – 07H
0	1	1	08H – 0FH
1	0	2	10H – 17H
1	1	3	18H – 1FH

Table No 05: Bit-pattern of Register Banks of PSW

V. PSW.2(OV): overflow flag

- This flag is set whenever the result of the signed number operation is too large, causing the high-order bit to overflow into the sign bit.
- The overflow flag is only used to detect errors in signed arithmetic operation.

VI. PSW.1: Unused bit.

VII. PSW.0(P): Parity flag

- The parity flag reflects the number of 1"s in the A register only.
- If the A register contains odd number of 1"s then P=1.
- If the register contains even number of 1"s then P=0.

4.1.2.7.2 Discussion of Memory Organization of Intel 8051

The 8051 Microcontroller has 128 bytes of Internal RAM and 4KB of on chip ROM. The RAM is also known as Data Memory and ROM is known as program memory /code memory. This code memory holds the actual 8051 program that is to be executed. In 8051 this memory is limited to 64K.

➢ Internal RAM of 8051:

Internal RAM is on-chip on the 8051. It is the fastest RAM available and the most flexible in terms of reading, writing, and modifying its content. The 128 bytes of RAM is organized as below:

1) Four register banks (Bank0, Bank1, Bank2, and Bank3) each of 8-bits. The by default bank register is Bank0. The remaining banks are selected with the help of RS0 and RS1 bits of PSW Register.
2) 16 bytes of bit addressable area and 80 bytes of general-purpose area (Scratch pad memory)

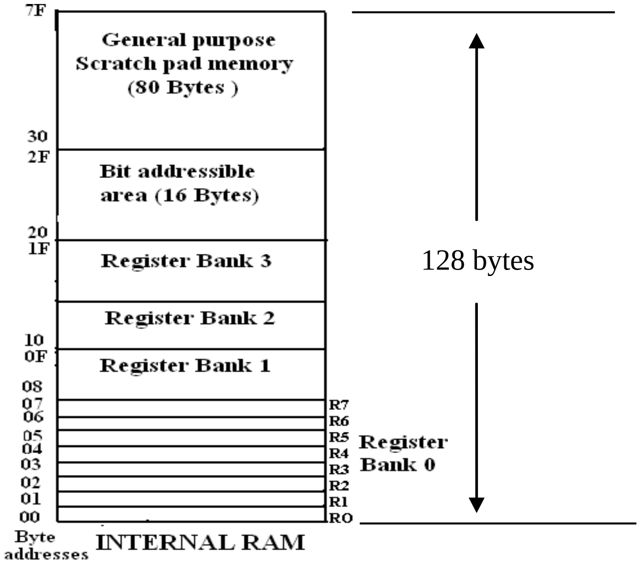


Figure 05: Memory Map of Internal RAM of 8051

➢ Internal ROM of 8051:

The 8051 microcontroller has 4KB of on-chip ROM but it can be extended up to 64KB. This ROM is also called as program memory or code memory. The code segment is accessed using the program counter (PC) for op-code fetches and by DPTR for data. The external ROM is accessed when the EA (active low) pin is connected to ground or the contents of program counter exceeds 0FFFH. When the internal ROM address is exceeded the 8051 automatically fetches the code bytes from the external program memory.

➢ Special function registers (SFR) of 8051:

In 8051 microcontrollers their certain registers which uses the RAM addresses from 80H to FFH and they are meant for certain specific operation. These registers are called special function register (SFR). Some of these registers are bit-addressable also.

The list of SFRs and their functional names are given below. In these SFRs some of them are related to I/O ports (P0, P1, P2 and P3) and some of them are meant for control operation (TCON, SCON, PCON) and remaining are the auxiliary SFRs in the sense that they don't directly configure the 8051.

Sr no.	Symbol		Name of SFRs	Address (Hex)
1.	ACC*		Accumulator	0E0H
2.	B*		B register	0F0H
3.	PSW*		Program status word register	0D0H
4.	SP		Stack pointer register	81H
5.	DPTR	DPL	Data pointer low byte	82H
		DPH	Data pointer high byte	83H
6.	P0*		Port 0	80H
7.	P1*		Port 1	90H
8.	P2*		Port 2	0AH
9.	P3*		Port 3	0BH
10.	IP*		Interrupt priority control	0B8H
11.	IE*		Interrupt Enable control	0A8H
12.	TMOD		Timer mode register	89H
13.	TCON*		Timer control register	88H
14.	TH0		Timer 0 higher byte	8CH
15.	TL0		Timer 0 lower byte	8AH

16.	TH1	Timer 1 higher byte	8DH
17.	TL1	Timer 1 lower byte	8BH
18.	SCON*	Serial control register	98H
19.	SBUF	Serial buffer register	99H
20.	PCON	Power control register	87H

Table No 06: Special Function Registers (SFR) of Microcontroller 8051

- ## Stack in 8051 of 8051:

The stack is a part of RAM used by the CPU to store information temporarily. This information may be either data or an address. The CPU needs this storage area as there are only limited numbers of registers. The register used to access the stack is called the stack pointer which is an 8-bit register. There are two important instructions to handle this stack. One is the PUSH and other is the POP.

4.1.2.7.3 I/O ports of 8051

The 8051 microcontrollers have four parallel I/O ports each of 8-bits. So, it provides the user 32 I/O lines for connecting the microcontroller to the peripherals. The four Ports are P0 (Port 0), P1 (Ports 1), P2 (Port 2) and P3 (Port 3).

- **Port 0 (P0):** Port 0 is an 8-bit I/O port with dual purpose. If external memory is used, these port pins are used for the lower address byte address/data (AD0 – AD7), otherwise all bits of the ports are either input or output. Unlike other ports, Port 0 is not provided with pull-up registers internally, so for Port 0 pull-up registers of nearly 10k are to be connected externally as shown in fig

Dual role of Port 0: Port 0 can also be used as address/data bus (AD0 – AD7), allowing it to be used for both address and data. When connecting the 8051 to an external memory, port 0 provide both address and data. The 8051 multiplexes address and data through port 0 to save the pins. ALE indicates whether port 0 has address or data. When ALE = 0, it provides data D0 – D7 and when ALE = 1, it provides address.

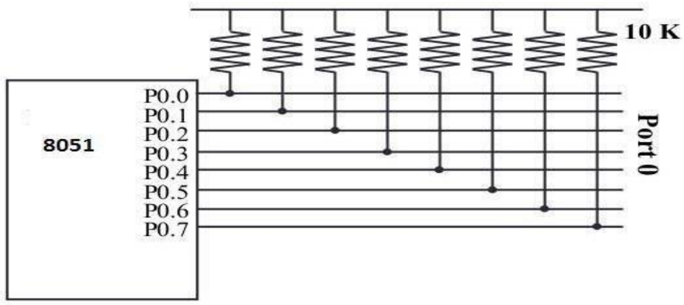


Figure 07: Circuit Diagram of Port 0

- **Port 1 (P1):** Port 1 occupies a total of 8 pins (pins 1 to 8). It has no dual application and act only as input or output port. In contrast to port 0, this port does not need any pull-up register since pull-up register connected internally.

- **Port 2 (P2):** Port 2 is also an 8-bit parallel port (pins 21 – 28). It can be used as input or output port. As this port is provided with internal pull-up resistor it does not need any external pull-up register.

  Dual role Port 2: P2 is used as simple I/O. In 8031 based system, port 2 must be used along with port 0 to provide the 16-bit address for external memory. Port 2 is also designated as A8 – A15, indicating its dual function. Since an 8051/8031 is capable of accessing 64K bytes of external memory, it needs a path for the 16-bits of the address. While P0 provides the lower 8-bits via A0 – A7, it is the job of P2 to provide bits A8 – A15 of the address. When the 8051/8031 is connected to external memory, P2 is used for the upper 8-bits of the 16-bit address, and it cannot be used for I/O.

- **Port 3 (P3):** Port 3 is also an 8-bit parallel Port with dual functions. The port pins can be used for I/O operations as well as for control operations. Port 3 does not need any external pull-up resistors as they are provided internally similar to the case of Port 2 &Port 1.

  Alternate functions of Port 3: P3.0 and P3.1 are used for the Receive Data (RXD) and Transmit Data (TXD) respectively. Bits P3.2 and P3.3 are meant for external interrupt. Bits P3.4 and P3.5 are used for Timer 0 and 1 and P3.6 and P3.7 are used to provide the write and read signal of external memories connected in 8031 based system.

4.1.2.7.4 Timers and Counters of 8051

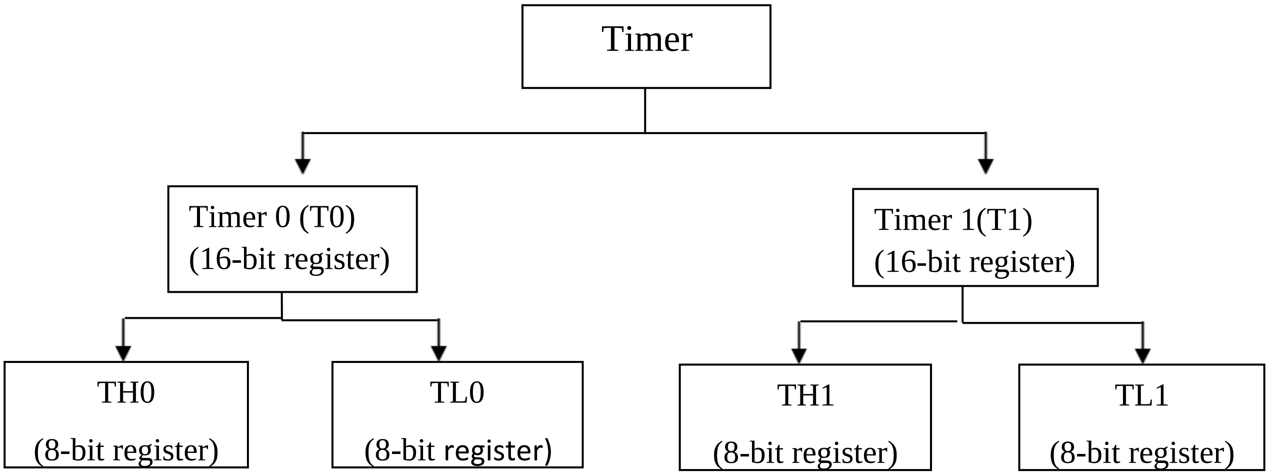


Figure No 08: Organization Chart of Timers of 8051

In the timer function mode, the counter is incremented in every machine cycle. Thus, one can think of it as counting machine cycles. Hence, the clock rate is 1/12th of the oscillator frequency.

In the counter function mode, the register is incremented in response to a 1 to 0 transition at its corresponding external input pin (T0 or T1). It requires 2 machine cycles to detect a high to low transition. Hence maximum count rate is 1/24th of oscillator frequency. The operation of the timers/counters is controlled by two special function register, TMOD and TCON respectively.

➢ Timer/Counter control logic of 8051:

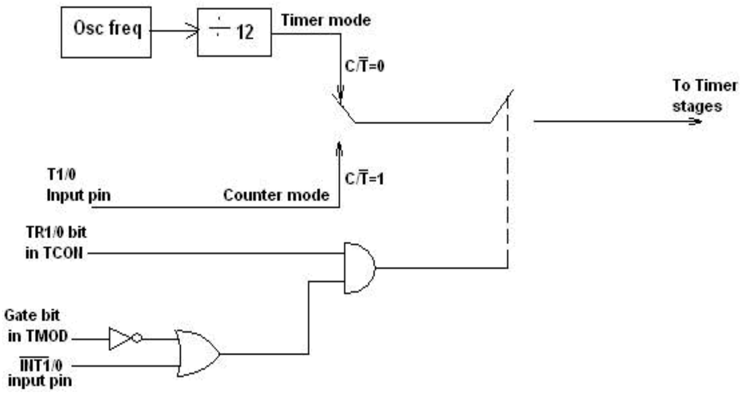


Figure 09: Control logic of Counter/Timer

➢ Timer Calculation:

F_{OSC} = 11.0592 MHz
Delay = 1 ms

Timer count can determine as below:

Tick= 1/(F_{OCS}/12)
Tick = 12/F_{OCS}

For F_{OCS}= 11.0592 MHz the tick time will be = 12/11.0592 MHz = 1.085µs
Timer value for the required can be calculated as below.

Count = Delay/ Tick = 1ms/1.085µs = 9216
Timer register = Max timer count – count

Timer register = 65536 – 9216 = 56320 = 0xDC00
TH = DC & TL = 00

➢ Timer Control (TCON) SFR of 8051:

D7	D6	D5	D4	D3	D2	D1	D0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Table No 07: Bit-pattern of TCON of microcontroller 8051

- **TF0/1(bit 5 & 7):** Timer overflow flag. It is set when timer rolls from all 1‟s to 0‟s. Cleared when processor vectors to execute ISR.
- **TR 0/1 (bit 4 & 6):** Timer run control bit. Set to 1 to start the timer /counter.
- **IE 0/1 (bit 1 &3):** External interrupt edge flag. Set by hardware when the external interrupt edge (H- to- L transition) is detected. It is cleared when interrupt is processed.
- **IT 0/1 (bit 0 & 2):** Interrupt type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupt.

➢ Timer Mode Control (TMOD) SFR of 8051:

GATE	C/T1	M1	M0	GATE	C/T0	M1	M0
Timer 1				Timer 0			

Table No 08: Bit-pattern of TMOD of 8051

- **GATE:** Gating control when set. The timer/counter is enabled only while the INTx pin is high and TRx control pin is set. When cleared, the timer is enabled whenever the TRx control bit is set.
- **C/T:** Timer or counter selected cleared for timer operation (input from internal system clock). Set for counter operation (input from Tx input pin).
- **M1:** Mode bit 1.

□ **M0:** Mode bit 0.

M1	M0	Mode	Operation Mode
0	0	0	13-bit Timer mode
0	1	1	16-bit timer mode
1	0	2	8-bit auto reload
1	1	3	Split timer mode

Table No 09: Bit-pattern of modes of TMOD of 8051

Mode 1: It is a 16-bit timer therefore it allows values from 0000H to FFFFH to be loaded into the timer's registers TL and TH. After TL and TH are loaded with a 16-bit initial value, the timer must be started. After the timer is started. It starts count up until it reaches its limit of FFFFH. When it rolls from FFFFH to 0000H, it sets a high a flag bit called timer flag (TF). After the timer reaches its limit and rolls over, in order to repeat the process, the registers TH and TL must be reloaded with the original value and TF must be reset to 0.

Mode 0: Mode 0 is exactly same like mode 1 except that it is a 13-bit timer instead of 16-bit. The 13-bit timer can hold values between 0000H to 1FFFH in TH and TL. Therefore, when timer reaches its maximum of 1FFFH, it rolls over to 0000H, and TF is raised.

Mode 2: It is an 8-bit timer that allows only values of 00H to FFH to be loaded into the timer's register TH. After TH is loaded with 8-bit value, the 8051 gives a copy of it to TL. Then the timer must be started. After timer is started, it starts to count up by incrementing the TL register. It counts up until it reaches its limit of FFH. When it rolls over from FFH to 00H. It sets high the Timer Flag (TF). When TI register rolls from FFH to 00H and TF is set to 1, TL is reloaded automatically with the original value kept by the TH register.

Mode 3: Mode 3 is also known as spilt timer mode. Timer 0 and 1 may be programmed to be in mode 0, 1 and 2 independently of similar mode for another timer. This is not true for mode 3; timers do not operate independently if mode 3 is chosen for timer 0. Placing timer 1 in mode 3 causes it to stop counting; the control bit TR1 and the timer 1 flag TF1are then used by timer 0.

4.1.2.7.5 Serial Data Input/output of 8051

The serial port of 8051 is full duplex, i.e., it can transmit and receive simultaneously.

The register SBUF is used to hold the data. The special function register SBUF is physically two registers. One is written only and is used to hold data to be transmitted out of the 8051 via TXD. The other is, read only and hold the received data from external sources via RXD. Both mutually exclusive registers have the same address 099H.

- **Serial Port Control (SCON) SFR of 8051:**

D7	D6	D5	D4	D3	D2	D1	D0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Table No 10: Bit-pattern of SCON of 8051

- **SM0/1 (Bits 7&6)**: To select the mode of operation.

SM0	SM1	Mode	Description
0	0	0	Shift register
0	1	1	8-bit UART
1	0	2	9-bit UART
1	1	3	9-bit UART

Table No 11: Bit-pattern of modes of operation of SCON

- **SM2(Bit 5):** Multiprocessor communication bit

  In mode 2 &3: Set/cleared by program to enable multiprocessor communication.
  1 – An interrupt is generated if bit 9 of a received data is 1.

  0 – No interrupt is generated if bit 9 of a received data is 0.

  In mode 1: No interrupt will be generated unless valid stop bit is received.
  In mode 0: Clear to 0.

- **REN (Bit 4):** Receive enable bit.
  1 – To enable reception.

  0 – To disable reception.

- **TB8 (Bit 3):** Transmit bit 8.

  Set/cleared by program in mode 2 & 3.

- **RB8 (Bit 2):** Receive bit 8.

  In mode 2 &3: Bit 8 of receive data.
  In mode 1: stop bit.

  In mode 0: not used.

- **TI (Bit 1):** Transmit interrupt flag.

  Set by hardware at the beginning of the stop bit in mode 1. Must be cleared by software.

- **RI (Bit 0):** Receive interrupt flag

  Set by hardware halfway through the stop bit time in mode 1. Must be cleared by software.

The Power Mode Control (PCON) SFR of 8051:

D7	D6	D5	D4	D3	D2	D1	D0
SMOD	X	X	X	GF1	GF0	PD	IDL

Table No 12: Bit-pattern of PCON of 8051

- **SMOD (Bit 7):** Serial mode bit used to determine the baud rate with timer 1.

$$\textit{Baud rate}= \frac{\text{oscillator frequency}}{\mathbf{N}(\mathbf{256}-\ \mathbf{TH1})}$$

If SMOD =0, then N=384 and if SMOD=1, then N=192.

- **GF1 & GF0 (Bit 3 & 2):** general purpose flags not implemented on the standard device.
- **PD (Bit 1):** This bit is the power down bit. Not implemented on the standard device.
- **IDL (Bit 0):** Activate the idle mode to save power. Not implemented on the standard device.

Baud rate in SMOD = 0

When SMOD = 0, the 8051 divides 1/12th of the crystal frequency by 32 and uses that frequency for Timer 1 to set the baud rate. In the case of XTAL = 11.0592 MHz we have:

Machine cycle frequency = 11.0592 MHz / 12 = 921.6 kHz

921.6 kHz / 32 = 28,800 Hz since SMOD = 0

This is the frequency used by the timer 1 to set the baud rate.

Baud rate = 28,800/3 = 9,600.

Baud rate in SMOD = 1

With the fixed crystal frequency, we can double the baud rate by making SMOD = 1. When SMOD bit is set to 1, 1/12 of XTAL is divided by 16 and that is the frequency used by timer 1 to set the baud rate. In case of XTAL = 11.0592 MHz, we have:

Machine cycle freq. = 11/0592 MHz/12 = 921.6 KHz and

921.6 KHz / 16 = 57,600 Hz since SMOD =1.

This is the frequency is used by the timer 1 to set the baud rate.

Baud rate = 57,600 / 3 = 19,200.

➢ Data Transmission

- Data transmission begins at any time when data is written in SBUF.
- TXD (P3.1) is used to transmit data to the serial data network.
- TI is set to 1 when data has been transmitted. This signifies that SBUF is empty so that another byte can be transmitted.

➢ Data Reception

- Reception of serial data begins if they receive enable bit is set 1 for all modes.
- RXD (P3.0) is used to receive data from the serial data network.
- Receive interrupt flag RI, is set after the data has been received in all modes. The data gets stored in SBUF register from where it can be read.

➢ Serial Data Transmission Mode of 8051

8051 Microcontroller communicate with another peripheral device through RXD and TXD pin of Port 3. Microcontrollers have four modes of serial communication as shown below:

1. Serial data mode 0 – Shift register
2. Serial data mode 1 – Standard UART
3. Serial data mode 2 – Multiprocessor (fixed baud rate)
4. Serial data mode 3 – Multiprocessor (variable baud rate)

1. Serial data mode 0 – Shift register (Fixed baud rate)

- In this mode, the Serial ports woks like a shift register and the data transmission works synchronously with a clock frequency of Fosc /12.
- Serial data is received and transmitted through RXD. 8-bits are transmitted /received at a time.
- Pin TXD outputs the shift clock pulses of frequency fosc /12, Which is connected to the external circuitry for synchronization.
- The shift frequency or baud rate is always 1/12th of the oscillator frequency.
- When transmitting, data is shifted out of RXD; the data changes on the falling edge of S6P2, or one clock pulse after the rising edge of the output TXD shift clock. The system designer must

design the external circuitry that receives this transmitted data to receive the data reliably based on this timing.

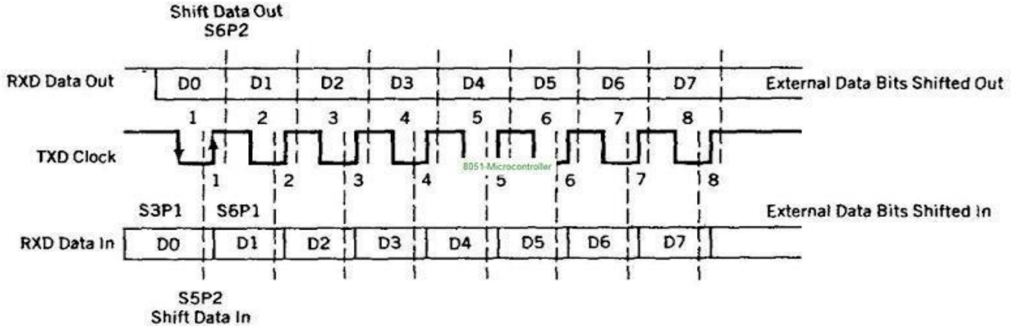


Figure No 10: Timing diagram of Serial data mode 0 of 8051

- Received data comes in on pin RXD and should be synchronized with the shift clock produced at TXD. Data is sampled on the falling edge of S5P2 and shifted in to SBUF on the rising edge of the shift clock.
- Mode 0 is not intended for data communication between computers, but as a high sped serial data-collection method using discreet logic to achieve high data rates.
- The baud rate used in mode 0 will be much higher than standard for any reasonable oscillator frequency; for a 6 megahertz crystal, the shift rate will be 500 kilohertz.

2. Serial data mode 1 – Standard UART (Variable baud rate)

- In mode 1, the serial port functions as Standard universal Asynchronous Receiver Transmitter mode. 10 bits are transmitted through TXD or received through RXD.
- 10 bits consists of one bit start bit (usually 0), 8 data bits and one stop bit (usually 1), once received the stop bit goes into RB8 in SFR SCON.
- The baud rate is variable.

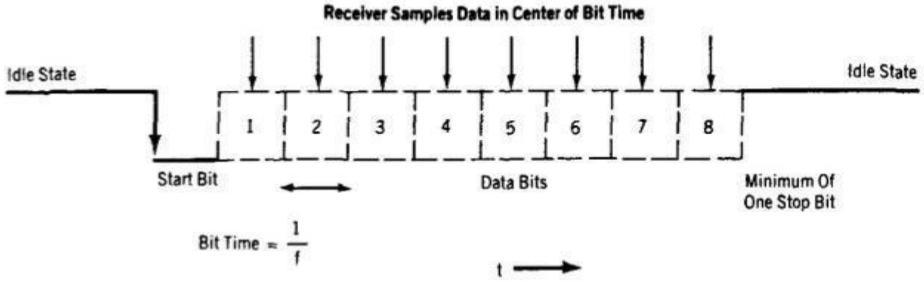


Figure No 11: Timing diagram of Serial data mode 1 of 8051

- Bit time = $1/f_{baud}$
- In receiving mode, data bits are shifted into the receiver at the programmed baud rate. The data word (8-bits) will be loaded to SBUF if the following conditions are true:

  (a) RI must be zero. (i.e. the previously received byte has been cleared from SBUF)

  (b) Mode bit SM2 = 0 or stop bit = 1.

Mode 1 baud rate generation:

Timer 1 is used to generate baud rate for mode 1 serial communication by using overflow flag of the timer to determine the baud frequency. Timer 1 is used in timer mode 2 as an auto reload 8-bit timer. The data rate is generated by timer 1 using the following formula:

$$\textbf{Baud rate} = \left(\frac{2^{SMOD1}}{32}\right) \times \left(\frac{SYSCLK \times 12^{(T1M-1)}}{(256 - TH1)}\right)$$

SMOD is the 7th bit of PCON register

F_{osc} is the crystal oscillator frequency of the microcontroller

It can be noted that f_{osc} / (12 x[256-(TH1)]) is the timer overflow frequency in timer mode 2, which is the auto-reload mode.

If timer-1 is not run in mode-2, then the baud rate is,

$Fbaud = 2SMOD32 \times (timer\ 1\ overflow\ frequency)$

Timer-1 can be run using the internal clock, fosc/12 (timer mode) or from any external source via pin T1 (P3.5) (counter mode).

3. Serial data mode 2 –Multiprocessor (Fixed baud rate)

- In this mode 11 bits are transmitted through TXD and receive through RXD.
- The various bits are as follows: a start bit (usually 0), 8 data bits (LSB First), a programmable 9th (TB8 or RB8) bit and a stop bit (usually 1).

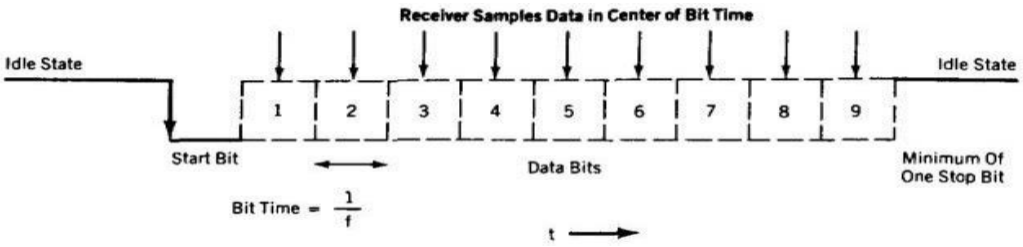


Figure No 12: Timing diagram of Serial data mode 2 of 8051

- While transmitting, the 9th data bit (TB8 in SCON) can be assigned the value „0‟ or „1‟. For example, if the information of parity is to be transmitted, the parity bit (P) in PSW could be moved into TB8.
- On reception of the data, the 9th bit goes into RB8 in „SCON‟, while the stop bit is ignored.
- The baud rate is programmable to either 1/32 or 1/64 of the oscillator frequency.

$$F_{baud} = 2_{SMOD}64d \times oscillatorfrequency$$

4. Serial data mode 3 – Multiprocessor (variable baud rate)

- In this mode 11 bits are transmitted through TXD and received through RXD.
- The various bits are: a start bit (usually 0), 8 data bits (LSB First), a programmable 9th bit and a stop bit (usually 1).
- Mode 3 is same as mode 2, except the fact that the baud rate in mode 3 is variable (i.e. just as in mode 1).

$$F_{baud} = 2_{SMOD}32 \times f_{osc}12[256 - TH1]$$

4.1.2.7.6 Interrupts of 8051

An interrupt is an external or internal event that disturbs the microcontroller to inform it that a device needs its service. The program which is associated with the interrupt is called interrupt service routine (ISR) or interrupt handler. Upon receiving the interrupt signal the microcontroller, finish current instruction and saves the PC on stack. Jump to a fixed location in memory depending on type of interrupt starts to execute the interrupt service routine until RETI (return from interrupt) upon executing the RETI the microcontroller returns to the place where it was interrupted.

Microcontroller 8051 have five Interrupts in addition to Reset. They are:

- Timer 0 overflow interrupt
- Timer 1 overflow interrupt
- External interrupt 0(INT0)
- External interrupt 1(INT1)
- Serial port interrupt

Each interrupt has a specific place in code memory where program execution (interrupt service routine) begins.

- External interrupt 0(INT0): 0003 H
- Timer 0 overflow interrupt: 000B H
- External interrupt 1(INT1): 0013 H
- Timer 1 overflow interrupt: 001B H
- Serial port interrupt : 0023 H

➢ Timer 0 and 1 Flag interrupt:

The timer flag is raised, when the timer rolls over. If the timer interrupt in the IE register is enabled, whenever, the timer rolls over, TF is raised, and the microcontroller is interrupted in whatever it is doing, and jumps to the interrupt vector table to service the ISR.

➢ External interrupt (INT0 and INT1):

Pin INTX are used by external circuitry. Inputs on these pins can set the interrupt flags IEX in the TCON register to 1. Bits ITX in TCON program the INTX pins for low-level interrupt when reset and program the INTX pins for high-to-low transition interrupt when set. Flags IEX will be reset when the processor accepts a transition-generated interrupt and the interrupt subroutine are accessed.

➢ Serial port interrupt (TI and RI):

Data byte is received or transmitted an interrupt bit; RI or TI set to 1 respectively in the SCON register. These are OR together to provide single interrupt to the processor, the serial port interrupt that is RI or TI. These bits are not cleared when the processor makes the interrupt-generated program call. The program that handles serial data communication must reset RI or TI to 0 to enable the next data communication operation.

➢ Interrupt Enable (IE) Register:

This register is responsible for enabling and disabling the interrupt. It is bit addressable register in which EA must be set to 1 for enabling the interrupt. The corresponding bits in this register enables particular interrupts like timer, external and serial inputs. In the below IE register, bit corresponding to 1 activates the interrupt and 0 disables the interrupts.

IE.7	IE.6	IE.5	IE.4	IE.3	IE.2	IE.1	IE.0
EA	-	-	ES	ET1	EX1	ET0	EX0

Table No 13: Bit-pattern of IE register of 8051

EA	IE.7	Disables all interrupt. If EA=0, no interrupt will be acknowledged. If EA=1, interrupt source is individually enabled or disabled by clearing it enable bit.
-	IE.6	Not implemented, reserved for future use.
-	IE.5	Not implemented, reserved for future use.
ES	IE.4	Enable or disable the serial port interrupt.
ET1	IE.3	Enable or disable the timer 1 overflow interrupt.
EX1	IE.2	Enable or disable the external interrupt (INT1).
ET0	IE.1	Enable or disable the timer 0 overflow interrupt.
EX0	IE.0	Enable or disable the external interrupt (INT0).

➢ Interrupt priority Register (IP):

It is also possible to change the priority levels of the interrupts by setting or clearing the corresponding bit in the interrupt priority (IP) register. This allows the low priority interrupt to interrupt the high priority interrupt, but prohibits the interruption by another low priority interrupt. Similarly, the high priority interrupt cannot be interrupted. If these interrupt priorities are not programmed, the microcontroller executes in predefined manner and its order is INT0, IF0, INT1, IF1 and SI.

IP7	IP6	IP5	IP4	IP3	IP2	IP1	IP0
-	-	PT2	PS	PT1	PX1	PT0	PX0

Table No 14: Bit-pattern of IP Register of 8051

-	IP7	Reserved
-	IP6	Reserved
PT2	IP5	Timer 2 interrupt priority bit (8052 only)
PS	IP4	Serial port interrupt priority bit
PT1	IP3	Timer 1 interrupt priority bit
PX1	IP2	External interrupt 1 priority bit
PT0	IP1	Timer 0 interrupt priority bit
PX0	IP0	External interrupt 0 priority bit

Priority bit = 1 assign high priority.
Priority bit = 0 assign low priority.

➢ Reset:

Reset occurs when the RS pin is supplied with a positive pulse in duration of at least 2 machine cycles. After that the microcontroller generates an internal reset signal Which clears all SFRs, except SBUF register, stack pointers and ports (the state of the first two ports are not defined, while FF value is written to the ports configuring all their pins as inputs). Depending on surrounding and purpose of device, the RS pin is usually connected to a power-on reset push button or circuit or to both of them. Reset is a non-makeable interrupt. A reset is accomplished by holding the RST pin high for at least 2 machine cycles. On resetting the program starts from 0000H and some flags are modified as follows:

Reset	**Value (hex) on reset**
PC	0000H
DPTR	0000H

A	00H
B	00H
SP	07H
PSW	00H
Port 0 – 3 latches	FFH
IP	XXX 00000 b
IE	0XX 00000 b
TCON	00H
TMOD	00H
TH0	00H
TL0	00H
TH1	00H
TL1	00H
SCON	00H
SBUF	XX H
PCON	0 XXXX XXX b

Table No 15: Value on Reset

4.1.2.7.7 Oscillator circuit (XTAL1 and XTAL2) of 8051:

Any microcontroller requires circuitry that generates the clock pulses by which all internal operations are synchronized.

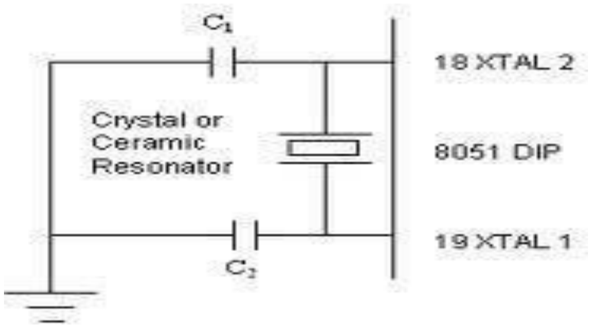


Figure 13: Internal circuit diagram of Oscillator

In AT8951 two pins viz, pin no. 18 & 19(XTAL1 & XTAL2) are provided for connecting a resonant network to form an oscillator. A quartz crystal is used with ceramic capacitors as shown in above circuit diagram. The crystal frequency is the basic internal frequency of the microcontroller.

The oscillator circuit usually runs around 12 MHz, although the 8051is capable of running at a maximum of 40 MHz (depending on specific model). Each machine cycle in the 8051 is 12 clock cycles, giving an effective cycle rate at 1 MHz (for a 12 MHz clock) and 3.33 MHz (for the maximum 40 MHz clock). The oscillator circuit generates the clock pulses so that all the internal operations are synchronized.

One machine cycle has 6 states. One state is 2 T-states. Therefore, one machine cycle is 12 T-states. Time to execute an instruction is found by multiplying C by 2 and dividing product by crystal frequency.

$$\boldsymbol{T}=\frac{(\mathrm{C}\times 12\mathrm{d})}{\mathrm{crystalfrequency}}$$

4.1.2.7.8 Addressing Modes of Microcontroller 8051

Addressing modes is an approach in which operands are given an instruction for further operation in an assembly language code of most of the CPU with an instruction set. 8051 supports 5 addressing modes. Types of addressing modes:

1. **Direct Addressing Mode:** In this addressing mode, the source and destination could be a register or RAM location, but both cannot be the same, either the source has to be a register followed by a RAM location as destination and vice versa. The address of the operand is specified in the instruction set itself. Internal RAM addresses starting from location 00H to 7FH and SFR addresses starting from 80H to FFH are only allowed in direct addressing mode.
2. **Register Addressing Mode:** In register addressing mode, the source and the destination both are registers, and must be the same size as indifference in size will give error. The data is specified in the register for various operations as per the given instructions. Only the Accumulator and R0 to R7 registers of each memory bank are allowed in this mode to transfer the data. The data transfer take place between Rn (R0 to R7) registers and the Accumulator(A) only and cannot be done between Rn register.
3. **Immediate Addressing Mode:** In immediate addressing mode, the data is directly specified in the instruction set itself. The source is the immediate data and the destination of this instruction could be any register. **'#'** symbol specifies that the operand is an immediate data on which operation is to be performed and the result will be stored in the destination register.
4. **Indirect Addressing Mode:** In indirect addressing mode, the address of the operand is specified in a register. Only the registers R0 to R1 are data pointers i.e. the data is stored on the RAM location is held by either R0 or R1. It can only use addresses from 00H to 7FH. „@" sign is used in the instruction and is placed before the registers R0 and R1 to make the two registers as pointers.

5. **Indexed Addressing Mode:** In indexed addressing mode, address is indirectly specified in Accumulator (A), Data Pointer (DPTR) and program counter (PC). It is usually the sum of the addresses stored at [A + DPTR] or [A + PC]. This addressing mode is very useful because ROM contains permanent data which is stored in the form of look-up tables. To access the look-up, the addresses are given as SUM of contents of two registers, where one acts as the base and other acts as the index within the table. Also, this mode is used to access data from the code memory and is denoted by „C‟ in the instruction.

4.1.2.7.9 C program using 8051 microcontrollers

- Write a C program that toggles only bit P1^6.

```
❼ #include<stdio.h>
sbit P1^6 = LED;

void MSDelay (unsigned int);
void main()

{

While (1)

{

P1^6 = 1;

Delay ();
P1^6 = 0;

}

}

void MSDelay (unsigned int itime)

{

Unsigned int i, j;

for (i = 0; i<= itime; i++)

for (j=0; j<1275; j++);

}
```

- Write a C program to generate 1KHz from c08051F020 IC interfacing DRV8313 to drive a 3 phase DC motor with 120 degree phase shift.

    -

```
//------------- PROGRAM FOR GENERATING A 3 PHASE SINGLE, EACH OF 120 DEGREE PHASE SHIFT FOR 1KHZ
      FREQUENCY-------------------------
//------------- SO WE HAVE TAKEN 3 TIMERS EACH OF 16 BIT AUTO-RELOAD MODE-------------------------------------------
      -------------
//------------- TIMER 2, 3 & 4 ARE IN USE WITH THE TERMINALS P2.1, P2.2 & P2.3---------------------------------------------
      ---

      ORG 0000H

      LJMP MAIN                  // JUMP TO MAIN

//------------- TIMER 2 ADDRESS..--------------------------------------------------------
//------------- SET P2.1 -----------------------------------------------------------------
//------------- LOAD TIMER 2--------------------------------------------------------------

       ORG 002BH                  // TIMER 2 SUBROUTINE
       CPL P2.1                // COMPLIMENT P2.1
       MOV TH2, #0FD65H            // LOAD TIMER 2
      MOV TL2, #0FD65H            // LOAD TIMER 2
      CLR TF2                  // CLEAR TIMER 2 OVERFLOW
      RETI                   // RETURN INTERRUPT

//--------------  DELAY TIME FOR PHASE 2----------------------------------------------------
DELAY:   MOV R0, #01B9H              // LOAD DIRRECT VALUES
LOOP:    DEC R0                   // DECREMENT IF NOT ZERO R0 RESISTER
       JNZ LOOP                  // JUMP IF NOT ZERO LOOP

//-------------  TIMER 3 ADDRESS..  // SET P2.2  // LOAD TIMER 3-----------------------------
      ORG 0073H                  // ADDRESS TO INTERREPT HANDLER OF TIMER 3 OVERLOAD
         CPL P2.2                 // SET P2.2
         MOV TMR3RLL, #0FD65H          // LOAD TIMER 3 L
      MOV TMR3RLH, #0FD65H          // LOAD TIMER 3 H
      CLR TF3                  // CLEAR TIMER 3 OVERFLOW
      SET TR3                  // SET TIMER 3 OVERFLOW

//------------  DELAY TIME FOR PHASE 3-----------------------------------------------------

DELAY1:  MOV R2, #0298H              // LOAD DIRECT VALUES
LOOP:    DEC R2                   // DECREMENT IF NOT ZERO TO R2 RESISTER
       JNZ LOOP                  // JUMP IF NOT ZERO LOOP

----------- // TIMER 4 ADDRESS.. // SET P2 // LOAD TIMER 4------------------------------------

       ORG 0083H                  // ADDRESS TO INTERREPT HANDLER OF TIMER 3 OVERLOAD
      CPL P2.3                 // COMPLIMENT P2.3
      MOV RCAP4L, #0FD65H           // LOAD TIMER 4 L
       MOV RCAP4H, #0FD65H           // LOAD TIMER 4 H
      CLR TF4                  // CLEAR TIMER 4 OVERFLOW
      SET TR4                  // SET TIMER 4 OVERFLOW

//------------- LOCATION OF MAIN SUBROUTINE-------------------------------------------------

       ORG 0100H

MAIN:   MOV OSCICN,#017H  // INTERNAL OSCILLATOR
```

```
//------------- ENABLE INTERNAL OSCILLATOR FREQUENCY FLAG ----------------------------------
//------------- ENABLE INTERNAL OSCILLATOR AS SYSTEM CLOCK ---------------------------------
//------------- INTERNAL OSCILLATOR ENABLE -----------------------------------------------
//------------- INTERNAL OSCILLATOR ENABLE, FREQUENCY IS 16MHZ -----------------------------

          MOV WDTCN.#0deH            // WATCHDOG TIMMER
          MOV WDTCM,#0adH            // WATCHDOG TIMMER
          MOV XBR1,#0AH            // ENABLE INTERRUPTS, TIMERS & SYSTEM CLOCK
          MOV XBR2,#040H           // ENABLE CROSSBAR

         MOV IE, #0AH              // ENABLE INTERRUPT AND TIMER 2
        MOV T2CON, #04H              // ENABLE TIMER 2 MODE 1 : 16 BIT AUTO RELOAD

           MOV EIE2, 05H             // ENABLE TIMER OVERFLOW 3 & 4

HERE:    JMP HERE                   // JUMP TO HERE

            END
```

Interfacing Diagram:

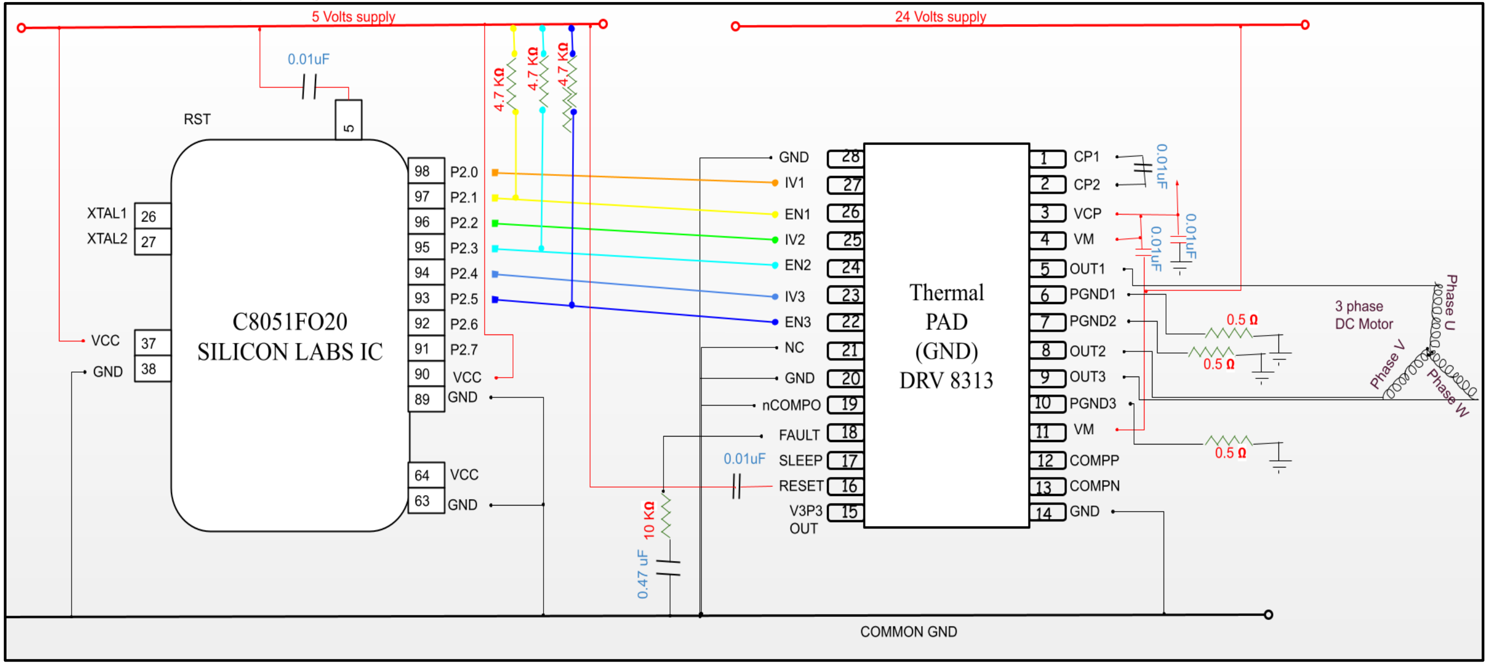


Figure: c08051f020 IC interfacing DRV8313 to drive a BLDC motor

4.2 Study of Silicon Laboratories C8051F020 (MDK)

4.2.1 Introduction

The C8051F020/1/2/3 devices are fully integrated mixed-signal System-on-a-chip MCUs with 64 digital I/O pins (C8051F020/2) or 32 digital I/O pins (C8051F021/3) introduced by silicon labs in 2003.

- High-speed pipelined 8051-compatible CIP-51 microcontroller core (up to 25 MIPS).
- In-system, Full-speed, non-instructive debug interface (on-chip).
- True 12-bir (C8051F020/1) or 10-bit (C8051F022/3) 100 ksps 8-channel ADC with PGA and analog multiplexer.

- True 8-bit ADC 500 ksps 8-channel ADC with PGA and Analog multiplexer.
- Two 8-bit ADC with programmable update scheduling.

- 64k bytes of in-system programmable FLASH memory.
- 4352 (4096+256) bytes of on-chip RAM.
- External data memory interface with 64k byte addresses space.
- SPI, SMBus/12C, and two UART serial interfaces implemented in hardware.
- Five general purpose 16-bit timers.
- Programmable Counter/Timer with five capture/compare modulus.
- On-chip Watchdog Timer, VDD Monitor and Temperature Sensor.

With On-chip VDD Monitor, Watchdog Timer and Clock Oscillator, the C8051F020/1/2/3 devices are truly stand-along System-on-a-chip solutions. All analog and digital peripherals are enabled/disabled and configured by user firmware. The FLASH memory can be reprogrammed even in-circuit, providing non-volatile data storage and also allowing field upgrades of the 8051 firmware.

On-board JTAG debug circuitry allows non-instructive (uses no 0n-chip resources) full speed, in-circuit debugging using the production MCU installed in the final application. This debug system supports inspection and modifications of memory and registers, setting breakpoints, watch points, single stepping, and run and halt commands. All analog and digital peripherals are fully functional while debugging using JTAG.

Each MCU is specified for 2.7 V-to-3.6 V operation over the industrial temperature range (-45° C to

+85° C). The port I/O, /RST and JTAG pins are tolerant for input signals up to 5 V. the C8051F020/2 is available in a 100-pin TQFP package.

- ➢ Supply Voltage:2.7 V to 3.6 V
  - Typical Operating Current: 10 mA @ 20 MHz
  - Multiple Power Saving Sleep and Shutdown Modes.
  - 100 pin TQFP and 64 pin TQFP package available.
  - Temperature range: -40°C to +85°C.
- ➢ High Speed 8051 µC Core
  - Pipelined Instruction Architecture; Executes 70%of Instruction Set in 1 or 2 system clocks.
  - Up to 25 MIPS Throughputs with 25 MHz clock
  - 22 Vectored Interrupt Sources.
- ➢ Clock sources
  - Internal Programmable Oscillator: 2 to 16 MHz
  - External Oscillator: Crystal, RC, C or clock

    - Real-Time Clock Mode using Timer-3 or PCA

- Memory
    - 4252 Bytes of Internal Data RAM (4k+256)
    - 64k Bytes FLASH; In system programmable in 512-bytes sectors
    - External 64k Bytes of Data Memory Interface (programmable multiplexed or non-multiplexed modes)
- Digital Peripherals
    - 8-Byte Wide port I/O (C8051F020/2); 5V tolerant
    - 4-Byte Wide port I/O (C8051F021/3); 5V tolerant
    - Hardware SMBus^{TM} (I^{2} C^{TM} Compatible), SPI^{TM} , and Two UART Serial Ports Available Concurrently.
    - Programmable 16-Bit Timer/Counter Array with 5 Capture/Compare modules
    - 5 General Purpose 16-Bit counter/Timers
    - Dedicated Watch-dog Timer; Bi-directional Reset pin
- Analog Peripherals
  - SAR ADC
    - 12-bit (C8051F020/1)
    - 10-bit (C8051F022/3)
    - ±1 LSB INL
    - Programmable Throughput up to 100 ksps
    - Up to 8 External Inputs; Programmable as Single-Ended or Differential
    - Programmable Amplifier Gain: 16, 8, 4, 2, 1, 0.5
    - Data-Dependent Windowed Interrupt Generator
    - Built-in Temperature Sensor (±3°C)
  - 8-Bit ADC
    - Programmable Throughput up to 500 ksps
    - 8 External Inputs
    - Programmable Amplifier Gain: 4, 2, 1, 0.5
  - Two 12-Bit DACs
    - Can Synchronize Outputs to Timers for Jitters-Free Wave form Generation

- Two Analog Comparators
- Voltage Reference
- Precision VDD Monitor/Brown Out Detector

- On-Chip JTAG Debug & Boundary Scan
  - On-Chip debug circuitry Facilitates Full-speed, Non-instructive In-Circuit/In-system Debugging
  - Provides Breakpoints, Single-Stepping, watch points, Stack monitor; Inspect/Modify Memory and Registers
  - Superior Performance to Emulations Systems Using ICE Chip, Target pods and Sockets
  - IEEE 1149.1 Compliant Boundary Scan
  - Low-cost Complete Development Kit

4.2.2 Pin Diagram of C8051F020

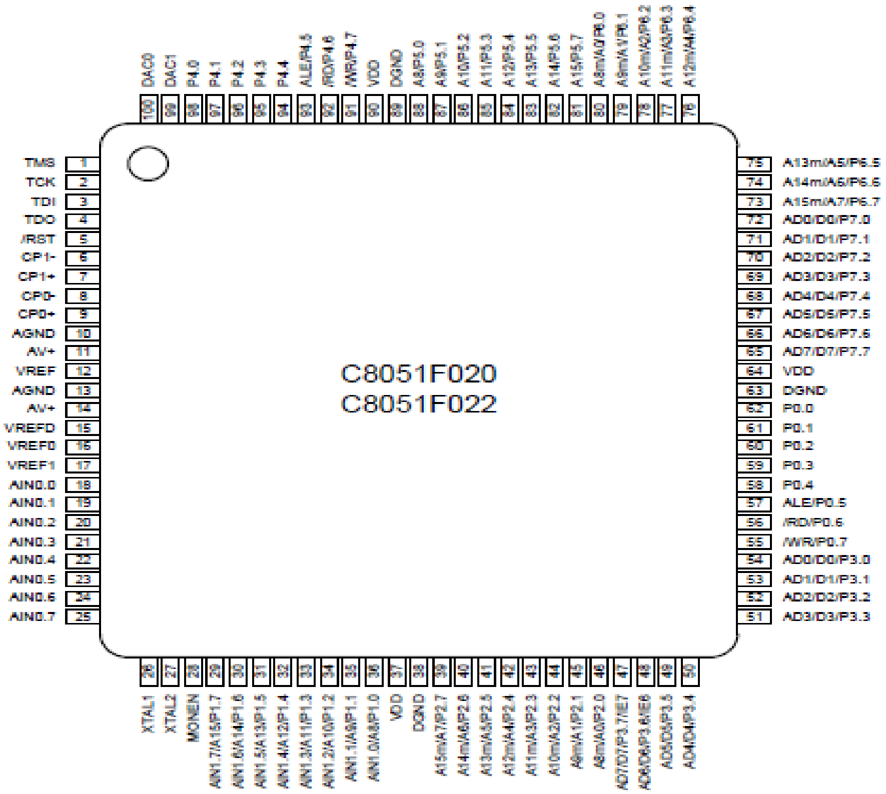


Figure 14: Pin Diagram of C8051F020

4.2.3 Block Diagram of C8051F020

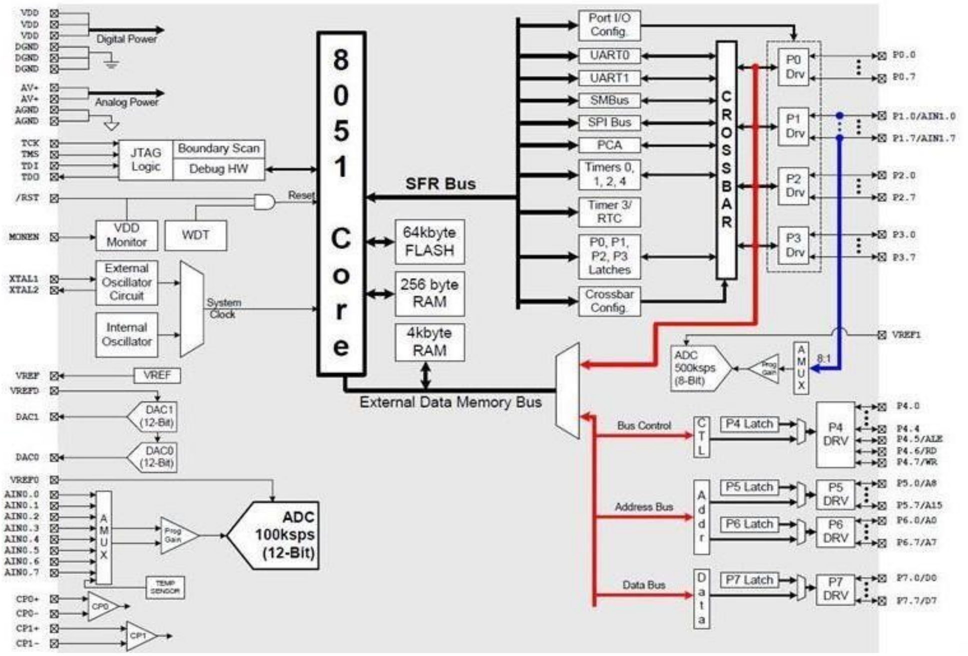


Figure 15: Block Diagram of C8051F020

4.2.4 ADC0 (12-Bit ADC) of C8051F020

The ADC0 Subsystem for the C8051F020 consist of a 9-channel, configurable analog multiplexed (AMUX0), a programmable gain amplifier (PGA0), and a 100 ksps, 12-bit successive approximation-register ADC with integrated Track-and-Hold and Programmable Window Detector. The AMUX0, PGA0, Data Conversion Modes and Window Detector are all configurable under software control via the Special Function Registers. The ADC0 subsystem (ADC0, tack-and-hold, PGA0) is enabled only when the AD0EN bit in the ADC0 Control register (ADC0CN) is set to logic 1. The ADC0 subsystem is in low power shutdown when this bit is logic 0.

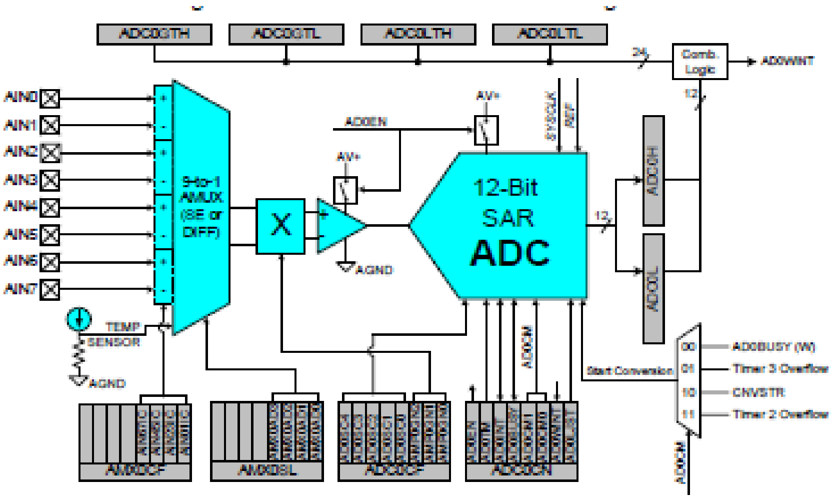


Figure 16: ADC0 Block Diagram of C8051F020

4.2.4.1 ADC Modes of Operation of C8051F020

ADC0 has a maximum conversion speed of 100 ksps. The ADC0 conversion clock is derived from the system clock divided by the value held in the ADCSC bits of register ADC0CF.

- **Starting a Conversion**

A conversion can be initiated in one of four ways, depending on the programmed states of the ADC0 Start of Conversion Mode bi (AD0CM1, AD0CM0) in ADC0CN. Conversions may be initiated by:

- Writing a „1" to AD0BUSY bit of ADC0CN;
- A Timer 3 overflow (i.e. timed continuous conversions);
- A rising edge detected on the external ADC convert start signal, CNVSTR;
- A timer „2" overflow (i.e. timed continuous conversions).

The AD0BUSY bit is set to logic 1 during conversion and restored to logic 0 when conversion is complete. The falling edge of AD0BUSY triggers an interrupt and sets the AD0INT interrupt flag (ADC0CN.5). Converted data is available in the ADC0 data word MSB and LSB registers, ADC0H, ADC0L.

- **Tracking Mode**

The AD0TM bit in register ADC0CN controls the ADC0 track-and-hold mode. In its default states, ADC0 input is continuously tracked when a conversion is not in progress. When the AD0TM bit is logic 1, ADC0 operates in low-power track-and-hold mode. In this mode, each conversion is preceded by a tracking period of SAR clock. When the CNVSTR signal is used to initiate conversions in low-power tracking mode, ADC0 tracks only when CNVSTR is low; conversion begins on the rising edge of CNVSTR.

4.2.5 Voltage Reference of C8051F020

The voltage reference circuit offers full flexibility in operating the ADC and DAC modules. Three voltage reference input pins allow each ADC and two DACs to reference an external voltage reference or the on-chip voltage reference output. ADC0 may also reference the DAC0 output internally, and ADC1 may reference the analog power supply voltage, via the VREF multiplexers.

The internal voltage reference circuit consists of a 1.2 V, 15 ppm/°C band gap voltage reference generator and a gain-of-two output buffer amplifier. The internal reference may be routed via the VREF pin to external system components or to the voltage reference input pins. Bypass capacitors of 0.1 µF and 4.7 µF are recommended from the VREF pin to AGND.

-	-	-	AD0VRS	AD1VRS	TEMPE	BIASE	REFBE
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Table No 16: REF0CN: Bit-pattern of Reference Control Register

Bit 7-5: UNUSED. Read = 000b; Write = don’t care

Bit 4: AD0VRS: ADC0 Voltage Reference Select
0: ADC0 voltage reference from VREF0 pin.
1: ADC0 voltage reference from DAC0 output

Bit 3: AD1VRS: ADC1 Voltage Reference Select
0: ADC1 voltage reference from VREF1 pin
1: ADC1 voltage reference from AV+

Bit 2: TEMPE: Temperature Sensor Enable Bit
0: Internal Temperature Sensor Off
1: Internal Temperature Sensor On

Bit 1: BIASE: ADC/DAC Bias Generator Enable Bit (Must be „1‟ if using ADC or DAC)
0: Internal Bias Generator Off
1: Internal Bias Generator On

Bit 0: REFBE: Internal Reference Buffer Enable Bit
0: Internal Reference Buffer Off
1: Internal Reference Buffer On. Internal voltage reference is driven on the VREF pin

4.2.6 CIP – 51 Microcontrollers

The CIP – 51 microcontroller core implements the standard 8051 organization and peripherals as well as additional custom peripherals and functions to extend its capability. The CIP – 51 includes the following features:

- Fully compatible with MCS – 51 instructions set.
- 25 MIPS Peak Throughput with 25 MHz clock
- 0 to 25 MHz Clock Frequency
- 256 Bytes of Internal RAM
- 8/4 Bytes-Wide I/O Ports
- Extended Interrupt Handler
- Reset Input
- Power Management Modes
- On-chip Debug Logic
- Program and Data Memory Security

4.2.7 Memory Organization of CIP – 51 Microcontrollers

The memory organization of CIP-51 System Controller is similarly to that of a standard 8051. There are two separate memory spaces: program memory and data memory. Program and data memory share the same address space but are accessed via different instruction types. There are 256 bytes of internal data memory and 64k bytes of internal program memory address space implemented within the CIP-51.

4.2.6.1.1 Program Memory of CIP – 51 Microcontrollers

The CIP-51 has a 64k byte program memory space. The MCU implements 65536 bytes of this program memory space as in-system re-programmed FLASH memory, organized in a contiguous block from addresses 0x0000 to 0xFFFF. 512 bytes (0x00EE to 0xFFFF) for this memory are reserved for factory use and not available for user program storage. Program memory is normally assumed to be read-only.

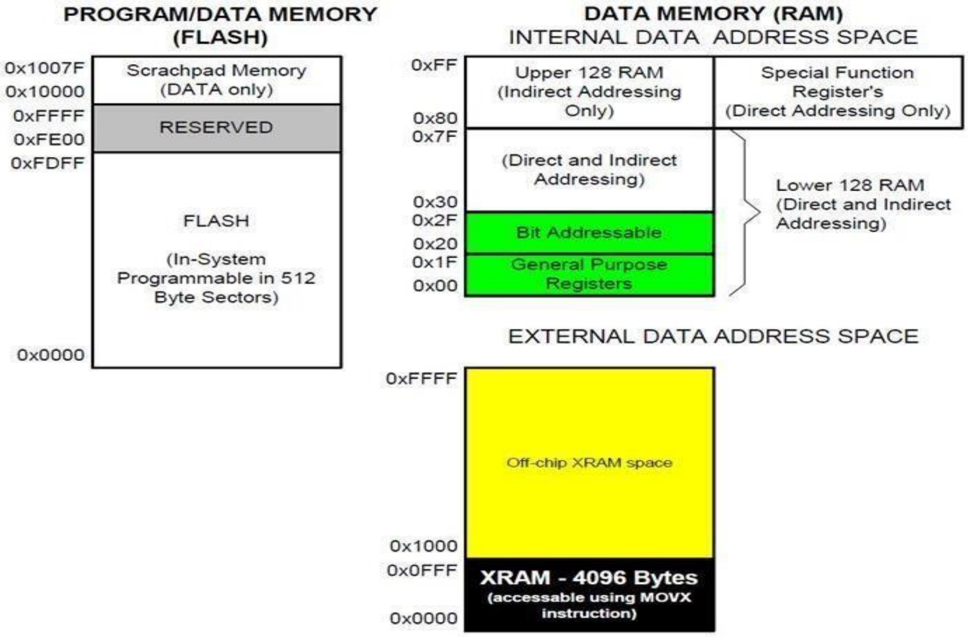

PROGRAM/DATA MEMORY (FLASH)
0x1007F
0x10000
Scratchpad Memory (DATA only)
0xFFFF
0xFE00
RESERVED
0xFDFF
FLASH
(In-System Programmable in 512 Byte Sectors)
0x0000
DATA MEMORY (RAM)
INTERNAL DATA ADDRESS SPACE
0xFF
Upper 128 RAM (Indirect Addressing Only)
Special Function Register's (Direct Addressing Only)
0x80
0x7F
(Direct and Indirect Addressing)
0x30
0x2F
Bit Addressable
0x20
0x1F
General Purpose Registers
0x00
Lower 128 RAM (Direct and Indirect Addressing)
EXTERNAL DATA ADDRESS SPACE
0xFFFF
Off-chip XRAM space
0x1000
0x0FFF
XRAM - 4096 Bytes
(accessable using MOVX instruction)
0x0000

Figure 17: Memory Organization of C8051F020

4.2.6.1.2 Data Memory of CIP – 51 Microcontrollers

The CIP-51 implements 256 bytes of internal RAM mapped into the data memory space from 0x00 through 0xFF. The 128 bytes of data memory are used for general purpose register and scratch pad. Either direct or indirect addressing may be used to access the lower 128 bytes of data memory. Locations 0x00 through 0x1F are addressable as four banks of general-purpose registers, each bank consisting of eight byte-wide registers. The next 16 bytes, locations 0x20 through 0x2F, may either be addressed as bytes or as 128-bit locations accessible with the direct addressing mode. The upper 128 bytes of data memory are accessible only by indirect addressing. This region occupies the same address space as the special function registers (SFR) but is physically separated from the space.

4.2.6.1.3 General Purpose Register of CIP – 51 Microcontrollers

The lower 32 bytes of data memory, locations 0x00 through 0x1F, may be addressed as four banks of general-purpose-registers. Each bank consists of eight byte-wide registers designated R0 through R7. Only one of these banks may be enabled at a time. Two bits in the program status word, RS0 (PSW.3) and RS1 (PSW.4), select the active register bank. Indirect addressing modes use register R0 and R1 as index register.

4.2.6.1.4 Stack of CIP – 51 Microcontrollers

A programmer’s stack can be located anywhere in the 256 byte data memory. The stack area is designated using the stack pointer (SP, address 0x81) SFR. The SP will point to the last location used. The next value pushed on the stack is placed at SP+1 and then SP is incremented.

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Table No 17: Bit-pattern of Stack Pointer (SP)

Bit 7-0: SP: Stack Pointer

The stack pointer holds the location of the top of the stack. The stack pointer is incremented before every PUSH operation. The SP registers defaults to 0x07 after reset.

4.2.6.1.5 Special Function Registers of CIP – 51 Microcontrollers

The direct-access data memory locations from 0x80 to FF constitute the special function registers (SFR). The SFR‟s provides the date exchange between the CIP-51‟s resources and the peripheral. The SFR registers are accessed anytime the direct addressing mode is used to access memory locations from 0x80 to 0xFF. SFR‟s with addresses ending in 0x0 or 0x8(e.g. P0, TCON, P1, SCON, IE, etc.) are bit-addressable as well as byte addressable only.

4.2.6.2 Interrupt Handler of CIP – 51 Microcontrollers

The CIP-51 includes an extended interrupt system supporting a total of 22 interrupt sources with two priority levels. Each interrupt sources have one or more associated interrupt-pending flag(s) coded in an SFR. When a peripheral or external source meets a valid interrupt condition, the associated interrupt pending flag is set to logic 1.

Each interrupt source can be individually enabled or disabled through the use of an associated interrupt enable bit in an SFR (IE-EIE2). However, interrupt must first be globally enabled by setting the EA bit (IE.7) to logic 1 before the individual interrupts are recognized. Setting the EA bit to logic 0 disables all interrupt sources regardless of the individual interrupt-enable setting.

External Interrupt: Two of the external interrupt sources (/INT0 and /INT1) are configurable as active-low level-sensitive or active-low edge-sensitive inputs depending on the setting of bits IT0 (TCON.0) and IT1 (TCON.2). IE0 (TCON.1) and IE1 (TCON.3) serve as the interrupt pending flag for the /INT0 and /INT1 external interrupts respectively.

Interrupt Priority: Each interrupt source can be individually programmed to one of the two priority levels: low or high. A low priority interrupt service routine can be preempted by a high priority interrupt. A high priority interrupt cannot be preempted. If two interrupts are recognized simultaneously, the interrupt with the higher priority is serviced first. If both interrupts have the same priority level, a fixed priority level is used.

4.2.7 Watchdog Timer Reset of CIP – 51 Microcontrollers

The MCU includes a programmable Watchdog Timer (WDT) running off the system clock. The WDT consist of 21- bit timer running from the programmed system clock. The timer measures the period between specific writes to its control register. If this period exceeds the programmed limit, a WDT reset is generated. The WDT can be enabled or disabled as needed in software, or can be permanently enabled

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 WDT control

Writing 0xA5 both enables and reloads the WDT.

Writing 0xDE followed within 4 system clocks by 0xAD disables the WDT.
Writing 0xFF locks out the disable feature.

Bit 4 Watchdog Status Bit (when read)
0: watchdog is inactive 1:
watchdog is active

Bit 2-0 Watchdog Timeout Interval bits

4.2.8. Oscillators of CIP – 51 Microcontrollers

Each MCU includes an internal oscillator and an external oscillator drive circuit, either of which can generate the system clock. The MCUs operate from the internal oscillator after any reset. The internal oscillator can be enabled/ disabled and its frequency can be set using the Internal Oscillator Control Register (OSCICN).

The MCUs can run from the internal oscillator permanently, or can switch to the external oscillator if desired using CLKSL bit in the OSCICN Register. The external oscillator requires an external resonator, crystal, capacitor, or RC network connected to the XTAL1/XTAL2 pins.

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MSCLKE	-	-	IFRDY	CLKSL	IOSCEN	IFCN1	IFCN0
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Figure 19: OSCICN: Bit-pattern of Internal Oscillator Control Register

Bit 7: MSCLKE: Missing Clock Enable Bit
0: Missing Clock Detector Disabled

1: Missing Clock Detector Enabled; reset triggered if clock is missing for more than 100µs

Bit 6-5: UNUSED: Read=00b, write=don't care

Bit 4: IFRDY: Internal Oscillator Frequency Ready Flag
0: Internal Oscillator Frequency not running

1: Internal Oscillator Frequency running

Bit 3: CLKSL: System Clock Source Select Bit
0: Uses Internal Oscillator as System Clock

1: Uses External Oscillator as System Clock

Bit 2: IOSCEN: Internal Oscillator Enable Bit
0: Internal Oscillator Disabled

1: Internal oscillator Enabled

Bit 1: IFCN1-0: Internal Oscillator Frequency Control Bits
00: Internal Oscillator Typical Frequency is 2 MHz
01: Internal Oscillator Typical Frequency is 4 MHz

10: Internal Oscillator Typical Frequency is 8 MHz
11: internal Oscillator Typical Frequency is 16 MHz

4.2.9 Port Input/output of CIP – 51 Microcontrollers

The C8051F020 are fully integrated mixed signal system on a chip MCUs with 64 digital I/O pins organized as 8-bit Ports. The lower ports: P0, P1, P2 and P3 are both bit and byte addressable through their corresponding port data register. The upper ports: P4, P5, P6 and P7 are byte addressable. All port pins are 5 V tolerant, and all support configurable open-drain or push-pull output modes and weak pull-ups.

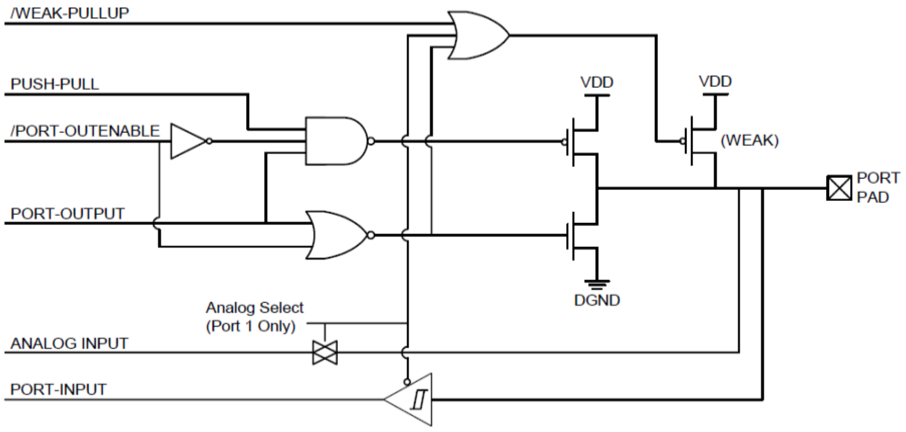


Figure 18: Internal circuit diagram of I/O Ports

The C8051F020 devices have a wide array of digital resources which are available through the four lower I/O ports: P0, P1, P2 and P3. Each of the pins on P0, P1, P2 and P3, can be defined as a General- purpose I/O (GPIO) pin or can be controlled by a digital peripheral or function. The ports pins of Port 1can be used as an Analog Inputs to ADC1.

4.2.9.1 Ports 0 through 3 of CIP – 51 Microcontrollers

The Priority Crossbar Decoder, or “Crossbar”, allocates and assigns Port pins on Port 0 through 3 to the digital peripherals on the device using a priority order. The ports pins are allocated in order starting with P0.0 and continue through P3.7. The digital peripherals are assigned Port pins in a priority order with UART0 having the highest order and CNVSTR having the lowest priority.

The crossbar assigns Port pins to a peripheral if the corresponding enable bits of the peripheral are set to logic 1in the crossbar configuration registers XBR0, XBR1 and XBR2. All Ports pins on ports 0 through 3 that are not allocated by the crossbar can be accessed as General-purpose I/O (GPIO) pins by reading and writing the associated port data registers, a set of SFRs which are both byte and bit addressable.

Weak Pull-Ups: By default, each port pin has an internal weak pull-up device enabled which provides a resistive connection between the pin and VDD. The weak-up devices can be globally disabled by writing a logic 1 to the Weak Pull-up Disabled Bit, (WEAKPUD, XBR2.7).

4.2.9.2 Ports 4 through 7 of CIP – 51 Microcontrollers

All Port pin on Ports 4 through 7 can be accessed as General purpose I/O (GPIO) pins by reading or writing the associated port data register, a set of SFRs which are byte addressable. A read of the port data register will always return the logic state present at the pin itself, regardless of whether the crossbar has allocated the pin for peripheral use or not.

The output mode of each port pin can be configured to be either Open-Drain or Push-Pull.

Weak Pull-Ups: By default, each port pin has an internal weak pull-up device enabled which provides a resistive connection between the pin and VDD. The weak pull-up devices can be globally disabled by writing a logical 1 to the Weak Pull-up Disabled Bit (WEAKPUD, XBR2.7). The Weak pull-up is automatically deactivated on any pin that is driving a logical 0; that is, an output pin will not contend with its own pull-up device.

4.2.10 UART0 of CIP – 51 Microcontrollers

UART0 is an enhanced serial port with frame error detection and address recognition hardware. UART0 may operate in full-duplex asynchronous or half-duplex synchronous mode. UART0 accessed via its associated SFRs, Serial Control (SCON0) and Serial Data Buffer (SBUF0). The Single SBUF0 location provide access to both transmit and receive registers. Read access the Receive register and Write access the Transmit register automatically.

UART0 has two sources of interrupt: A Transmit Interrupt Flag (TI0) set when transmission of a data byte is complete, and a Receive Interrupt Flag (RI0) set when reception of a data byte is complete. UART0 provides four operating modes (one synchronous and three asynchronous) selected by setting the configuration bits in the SCON0 register. These four modes offer different baud rate and communication protocols. The four modes are

Table 20: modes of operations of UART0

Modes	Synchronization	Baud Clock	Data Bits	Start/Stop Bits
0	synchronous	SYSCLK/12	8	None
1	Asynchronous	Timer 1 or 2 Overflow	8	1 Start, 1 Stop
2	Asynchronous	SYSCLK/32 or SYSCLK/64	9	1 Start, 1 Stop
3	Asynchronous	Timer 1 or 2 Overflow	9	1 Start, 1 Stop

4.2.11 Timers of CIP – 51 Microcontrollers

The C8051F020 devices contain 5 counter/timers: three 16-bit counter/timers and two 16-bit auto-reload timer for use with the ADCs, SMBus, UART1 or for the general-purpose use. These can be used to measure the time intervals, count external events and generate periodic interrupt requests. Timer 0 and timer 1 are nearly identical and have four primary modes of operation. Timer 2 offers additional capabilities not available in timer 1 and 0. Timer 3 is similar to timer 2, but without the capture or baud rate generator modes. Timer 4 is identical to timer 2 and can supply baud generation capabilities to UART1.

Table 21: Timers of C8051F020

Timer 0 and Timer 1	Timer 2	Timer 3	Timer 4
13 bit counter/timer	16 bit counter/timer with auto-reload	16 bit timer with auto-reload	16 bit counter/timer with auto-reload
16 bit counter/timer	16 bit counter/ timer with capture		16 bit counter/timer with capture
8 bit counter/timer with auto reload	Baud rate generator for UART0		Baud rate generator for UART1
Two 8 bit counter/timer (timers only)			

5. DC motor

A DC motor is an electrical machine that converts electrical energy into mechanical energy. In a DC motor, the input electrical energy is the direct current which is transformed into the mechanical rotation.
A DC motor is defined as a class of electrical motors that convert direct current electrical energy into mechanical energy.

From the above definition, we can conclude that any electric motor that is operated using direct current or DC is called a DC motor. We will understand the DC motor construction and how a DC motor converts the supplied DC electrical energy into mechanical energy in the next few sections.

DC Motor Parts

In this section, we will be discussing the construction of DC motors.

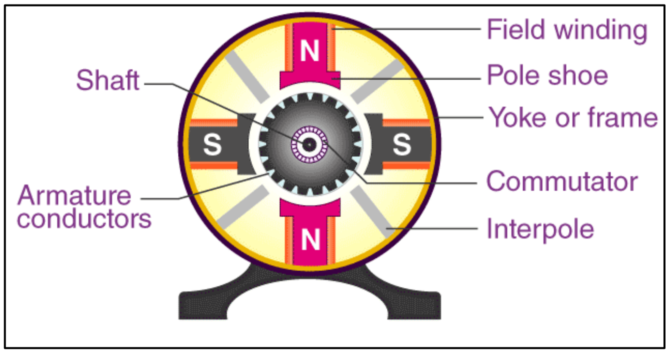


Figure : DC Motor construction parts Diagram

Different Parts of a DC motor

A DC motor is composed of the following main parts::

Armature or Rotor

The armature of a DC motor is a cylinder of magnetic laminations that are insulated from one another. The armature is perpendicular to the axis of the cylinder. The armature is a rotating part that rotates on its axis and is separated from the field coil by an air gap.

Field Coil or Stator

A DC motor field coil is a non-moving part on which winding is wound to produce a magnetic field. This electro-magnet has a cylindrical cavity between its poles.

Commutator and Brushes

Commutator

The commutator of a DC motor is a cylindrical structure that is made of copper segments stacked together but insulated from each other using mica. The primary function of a commutator is to supply electrical current to the armature winding.

Brushes

The brushes of a DC motor are made with graphite and carbon structure. These brushes conduct electric current from the external circuit to the rotating commutator. Hence, we come to understand that the commutator and the brush unit are concerned with transmitting the power from the static electrical circuit to the mechanically rotating region or the rotor**.**

DC Motor Working Explained

In the previous section, we discussed the various components of a DC motor. Now, using this knowledge let us understand the working of DC motors.

A magnetic field arises in the air gap when the field coil of the DC motor is energised. The created magnetic field is in the direction of the radii of the armature. The magnetic field enters the armature from the North pole side of the field coil and “exits” the armature from the field coil’s South pole side.

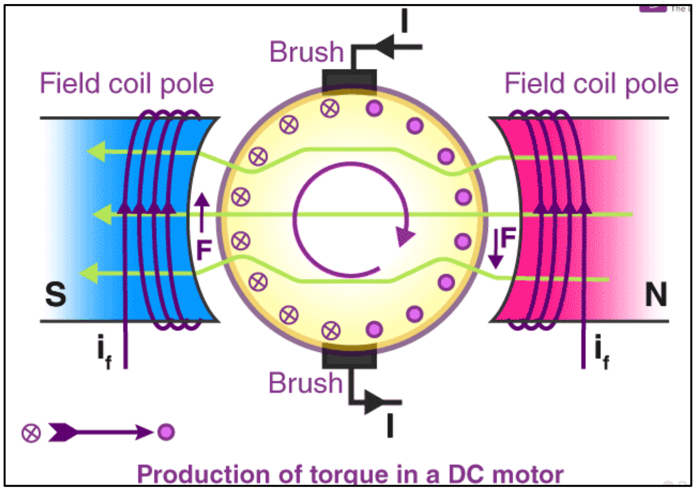
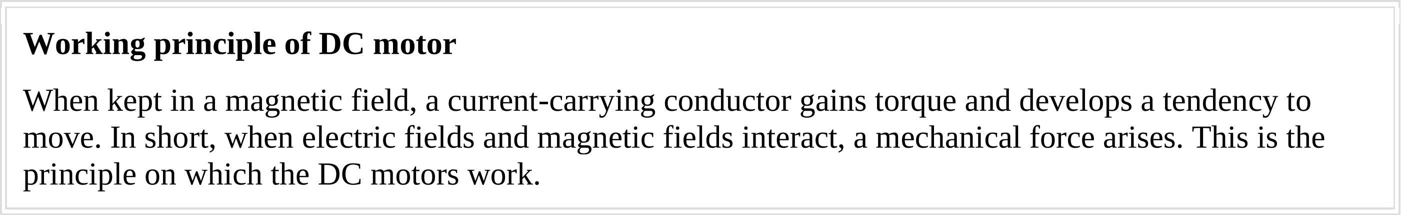


Figure : Production of torque in a DC motor

The conductors located on the other pole are subjected to a force of the same intensity but in the opposite direction. These two opposing forces create a torque that causes the motor armature to rotate.



Working principle of DC motor

When kept in a magnetic field, a current-carrying conductor gains torque and develops a tendency to move. In short, when electric fields and magnetic fields interact, a mechanical force arises. This is the principle on which the DC motors work.

Figure : On working principle of DC motor

Types of DC motor

DC motors have a wide range of applications ranging from electric shavers to automobiles. To cater to this wide range of applications, they are classified into different types based on the field winding connections to the armature as:

- Self Excited DC Motor
- Separately Excited DC Motor

Now, let us discuss the various types of DC Motors in detail.

Self Excited DC Motor

In self-excited DC motors, the field winding is connected either in series or parallel to the armature winding. Based on this, the self-excited DC motor can further be classified as:

- Shunt wound DC motor
- Series wound DC motor
- Compound wound DC motor

Shunt wound DC motor

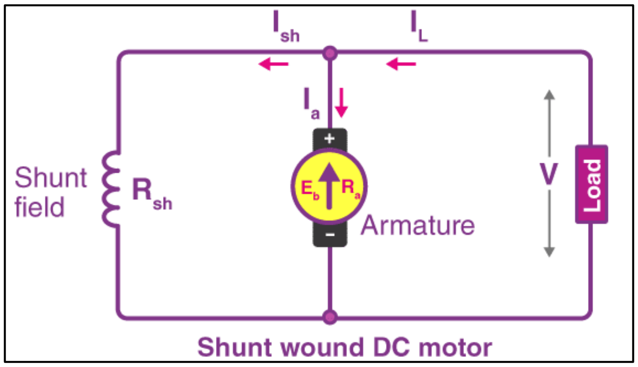


Figure : Shunt wound DC motor

In a shunt wound motor, the field winding is connected parallel to the armature as shown in the figure.

Series wound DC

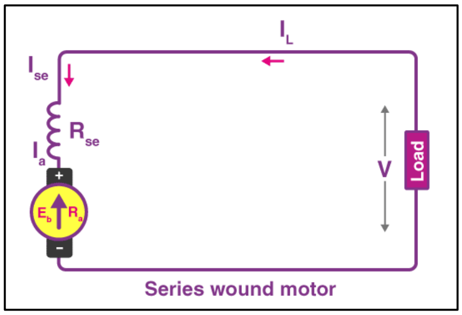
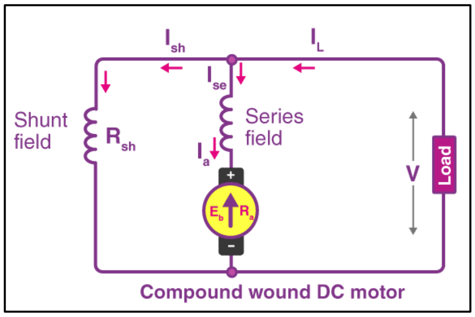


Figure : Series wound DC

In a series wound DC motor, the field winding is connected in series with the armature winding as shown in the figure.

Compound wound DC motor



DC motors having both shunt and series field winding is known as Compound DC motor, as shown in the figure. The compound motor is further divided into:

- Cumulative Compound Motor
- Differential Compound Motor

In a cumulative compound motor, the magnetic flux produced by both the windings is in the same direction. In a differential compound motor, the flux produced by the series field windings is opposite to the flux produced by the shunt field winding.

Separately Excited DC Motor

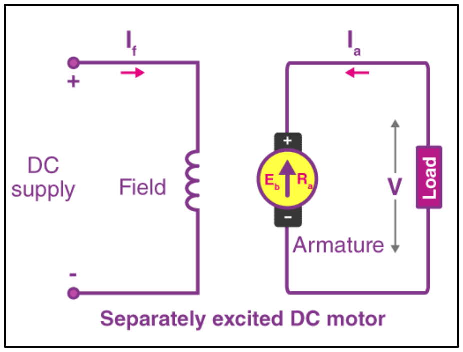


Figure : Separately Excited DC Motor

In a separately excited DC motor, the field coils are energised from an external source of DC supply as shown in the figure.

Brushed DC Motor vs Brushless DC Motor

A brushless DC motor, also known as synchronous DC motor, unlike brushed DC motors, do not have a commutator. The commutator in a brushless DC motor is replaced by an electronic servomechanism that can detect and adjust the angle of the rotor.

A brushed DC motor features a commutator that reverses the current every half cycle and creates single direction torque. While brushed DC motors remain popular, many have been phased out for more efficient brushless models in recent years.

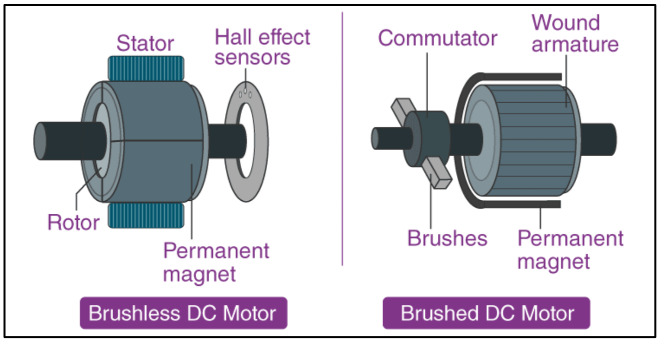


Figure : Brushed DC motor v/s Brushless DC motor

Applications of DC Motor

The applications of different types of DC motors are listed below:

Shunt DC Motors

Owing to the fairly constant speed and medium starting torque of shunt DC motors, they are used in the following applications:

1. Centrifugal and reciprocating pumps
2. Lathe machines
3. Blowers and Fans
4. Drilling machines
5. Milling machines
6. Machine tools

Series DC Motors

Owing to the high starting torque and variable speed of series DC motors, they are used in the following applications:

- Conveyors
- Hoists, Elevators
- Cranes
- Electric Locomotives

5.1 INTRODUCTION:

Pelletron accelerator produces ion beam of different ions. This ion beams are given to users for experiments. Operator takes the help of different diagnostic devices like BPM, Faraday cup, slits to tune the beam and DC motors.

➢ **Major Requirement**

1. Silicon Labs (C8051F020)
2. DRV8313 Triple half H Bridge Driver IC
3. DC MOTOR

5.2.Silicon Labs (C8051F020)

The silicon labs C8051F020 is advance version of microcontroller 8051. It consists of 100 pins. And it also internally consists of 16 on-chip ADC and 2 on-chip DAC and it consist of 8 I/O ports which we can use as a multiplex address or data bus. So, in slit Controller we are using ADC channels, I/O ports for LCD and Keypad and so many applications.

The C8051F02 Development Kit includes a target board with a C8051F020 device pre-installed for evaluation and preliminary software development. Numerous Input/output (I/O) connections are provided to facilitate prototyping using the target board.

- P1 Power connector (accepts input from 7 to 15 VDC unregulated power adapter)
- J1 Connects SW2 to P3.7 pin
- J3 Connects LED D3 to P1.6 pin
- J4 JTAG connector for Debug Adapter interface
- J5 DB-9 connector for UART0 RS232 interface
- J6 Connector for UART0 TX (P0.0)
- J8 Connector for UART0 RTS (P4.0)
- J9 Connector for UART0 RX (P0.1)
- J10 Connector for UART0 CTS (P4.1)
- J11 Analog loopback connector
- J12-J19 Port 0 - 7 connectors
- J20 Analog I/O terminal block
- J22 VREF connector
- J23 VDD Monitor Disable
- J24 96-pin Expansion I/O connector

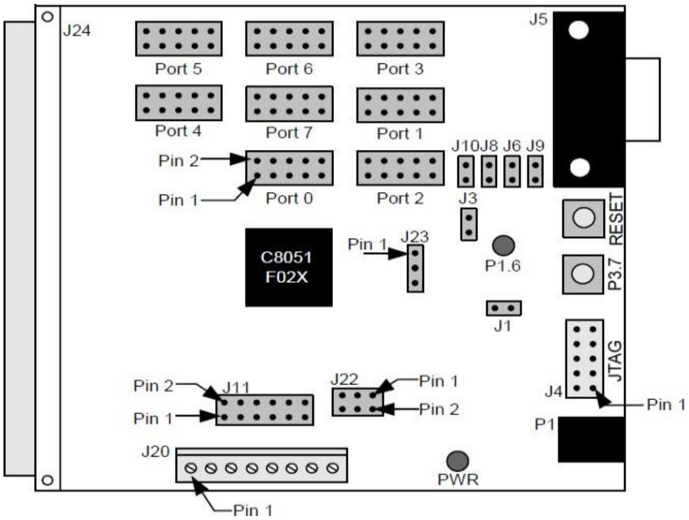


Figure 21: C8051F020 Target Board

➢ Timer

The C8051F020/1/2/3 devices contain 5 counter/timers: three are 16-bit counter/timers compatible with those found in the standard 8051, and two are 16-bit auto-reload timers for use with the ADCs, SMBus, UART1, or for general purpose use. These can be used to measure time intervals, count external events and generate periodic interrupt requests. Timer 0 and Timer 1 are nearly identical and have four primary modes of operation. Timer 2 offers addi- tional capabilities not available in Timers 0 and 1. Timer 3 is similar to Timer 2, but without the capture or Baud Rate Generator modes. Timer 4 is identical to Timer 2, and can supply baud-rate generation capabilities to UART1.

Timer 0 and Timer 1:	**Timer 2:**	**Timer 3:**	**Timer 4**
13-bit counter/timer	16-bit counter/timer with auto-reload	16-bit timer with auto-reload	16-bit counter/timer with auto-reload
16-bit counter/timer	16-bit counter/timer with capture		16-bit counter/timer with capture
8-bit counter/timer with auto-reload	Baud rate generator for UART0		Baud rate generator for UART1
Two 8-bit counter/timers (Timer 0 only)			

When functioning as a timer, the counter/timer registers are incremented on each clock tick. Clock ticks are derived from the system clock divided by either one or twelve as specified by the Timer Clock Select bits (T4M-T0M) in CKCON, shown in Figure 22.1. The twelve-clocks-per-tick option provides compatibility with the older generation of the 8051 family. Applications that require a faster timer can use the one-clock-per-tick option.

When functioning as a counter, a counter/timer register is incremented on each high-to-low transition at the selected input pin. Events with a frequency of up to one-fourth the system clock's frequency can be counted. The input signalneed not be periodic, but it should be held at a given level for at least two full system clock cycles to ensure the levelis sampled.

Timer 0 and Timer 1

Timer 0 and Timer 1 are accessed and controlled through SFRs. Each counter/timer is implemented as a 16-bit regis- ter accessed as two separate bytes: a low byte (TL0 or TL1) and a high byte (TH0 or TH1). The Counter/Timer Con- trol (TCON) register is used to enable Timer 0 and Timer 1 as well as indicate their status. Both counter/timers operate in one of four primary modes selected by setting the Mode Select bits M1-M0 in the Counter/Timer Mode (TMOD) register. Each timer can be configured independently. Following is a detailed description of each operating mode.

Timer 2

Timer 2 is a 16-bit counter/timer formed by the two 8-bit SFRs: TL2 (low byte) and TH2 (high byte). As with Timers 0 and 1, Timer 2 can use either the system clock or transitions on an external input pin (T2) as its clock source. The Counter/Timer Select bit C/T2 bit (T2CON.1) selects the clock source for Timer 2. Clearing C/T2 selectsthe system clock as the input for the timer (divided by either one or twelve as specified by the Timer Clock Select bitT2M in CKCON). When C/T2 is set to 1, high-to-low transitions at the T2 input pin increment the counter/timer register. for information on selecting and configuring external I/O pins for digital peripherals.) Timer 2 can also be used to start an ADC Data Conversion.

Timer 2 offers capabilities not found in Timer 0 and Timer 1. It operates in one of three modes: 16-

bit Counter/Timerwith Capture, 16-bit Counter/Timer with Auto-Reload or Baud Rate Generator Mode. Timer 2's operating mode is selected by setting configuration bits in the Timer 2 Control register (T2CON). Below is a summary of the Timer 2 operating modes and the T2CON bits used to configure the counter/timer. Detailed descriptions of each mode follow.

RCLK0	**TCLK0**	**CP/RL2**	**TR2**	**Mode**
0	0	1	1	16-bit Counter/Timer with Capture
0	0	0	1	16-bit Counter/Timer with Auto-Reload
0	1	X	1	Baud Rate Generator for UART0
1	0	X	1	Baud Rate Generator for UART0
1	1	X	1	Baud Rate Generator for UART0
X	X	X	0	Off

Mode 0: 16-bit Counter/Timer with Capture

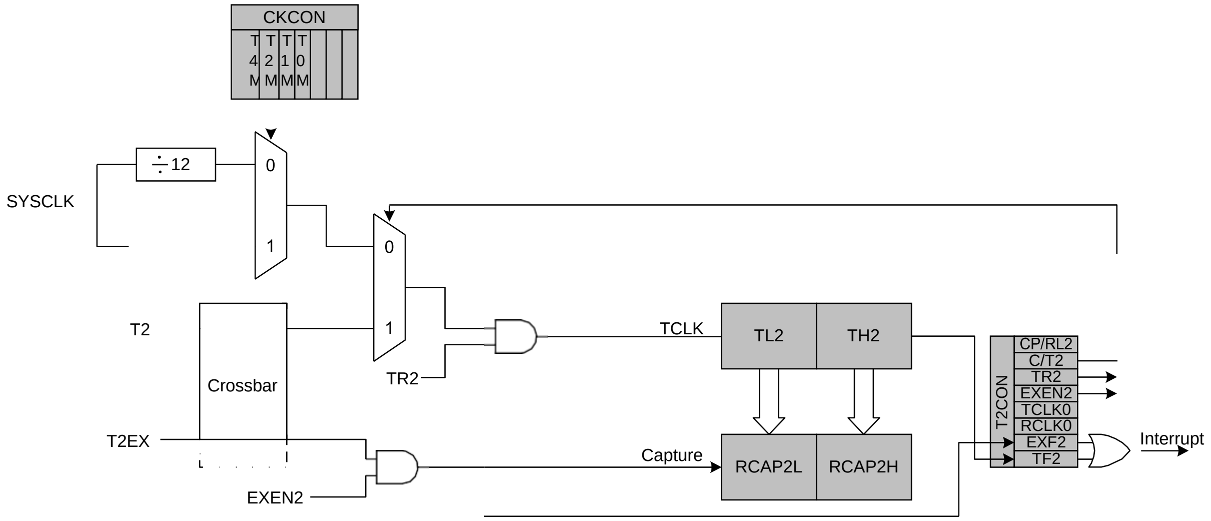
In this mode, Timer 2 operates as a 16-bit counter/timer with capture facility. A high-to-low transition on the T2EX input pin causes the following to occur:

1. The 16-bit value in Timer 2 (TH2, TL2) is loaded into the capture registers (RCAP2H, RCAP2L).
2. The Timer 2 External Flag (EXF2) is set to ‘1’.
3. A Timer 2 interrupt is generated if enabled.

Timer 2 can use either SYSCLK, SYSCLK divided by 12, or high-to-low transitions on the T2 input pin as its clock source when operating in Capture mode. Clearing the C/T2 bit (T2CON.1) selects the system clock as the input for the timer (divided by one or twelve as specified by the Timer Clock Select bit T2M in CKCON). When C/T2 is set tologic 1, a high-to-low transition at the T2 input pin increments the counter/timer register. As the 16-bit counter/timer register increments and overflows from 0xFFFF to 0x0000, the TF2 timer overflow flag (T2CON.7) is set and an interrupt will occur if the interrupt is enabled.

Counter/Timer with Capture mode is selected by setting the Capture/Reload Select bit CP/RL2 (T2CON.0) and the Timer 2 Run Control bit TR2 (T2CON.2) to logic 1. The Timer 2 External Enable EXEN2 (T2CON.3) must also be set to logic 1 to enable a capture. If EXEN2 is cleared, transitions on T2EX will be ignored.

T2 Mode 0 Block Diagram



T2CON: Timer 2 Control Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
TF2	EXF2	RCLK0	TCLK0	EXEN2	TR2	C/T2	CP/RL2	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0xC8

Bit7: TF2: Timer 2 Overflow Flag.

Set by hardware when Timer 2 overflows. When the Timer 2 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 2 interrupt service routine. This bit is not automatically clearedby hardware and must be cleared by software. TF2 will not be set when RCLK0 and/or TCLK0 are logic 1.

Bit6: EXF2: Timer 2 External Flag.

Set by hardware when either a capture or reload is caused by a high-to-low transition on the T2EX input pin and EXEN2 is logic 1. When the Timer 2 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 2 Interrupt service routine. This bit is not automatically cleared by hard-ware and must be cleared by software.

Bit5: RCLK0: Receive Clock Flag for UART0.

Selects which timer is used for the UART0 receive clock in modes 1 or 3.0: Timer 1 overflows used for receive clock.

1: Timer 2 overflows used

for receive clock.Bit4: TCLK0:

Transmit Clock Flag for UART0.

Selects which timer is used for the UART0 transmit clock in modes 1 or 3.0: Timer 1 overflows used for transmit clock.

1: Timer 2 overflows used for transmit clock.

Bit3: EXEN2: Timer 2 External Enable.

Enables high-to-low transitions on T2EX to trigger captures or reloads when Timer 2 is not operatingin Baud Rate Generator mode.

0: High-to-low transitions on T2EX ignored.

1: High-to-low transitions on T2EX cause a capture or reload.

Bit2: TR2: Timer 2 Run Control.

This bit enables/disables Timer 2.0: Timer 2 disabled.

1: Timer 2 enabled.

Bit1: C/T2: Counter/Timer Select.

0: Timer Function: Timer 2 incremented by clock defined by T2M (CKCON.5).

1: Counter Function: Timer 2 incremented by high-to-low transitions on external input pin (T2).

Bit0: CP/RL2: Capture/Reload Select.

This bit selects whether Timer 2 functions in capture or auto-reload mode. EXEN2 must be logic 1 forhigh-to-low transitions on T2EX to be recognized and used to trigger captures or reloads. If RCLK0or TCLK0 is set, this bit is ignored and Timer 2 will function in auto-reload mode.

0: Auto-reload on Timer 2 overflow or high-to-low transition at T2EX (EXEN2 = 1). 1: Capture on high-to-low transition at T2EX (EXEN2 = 1).

Timer 2 Capture Register Low Byte

RCAP2L: Timer 2 Capture Register Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xCA

Bits 7-0: RCAP2L: Timer 2 Capture Register Low Byte.

The RCAP2L register captures the low byte of Timer 2 when Timer 2 is configured in capture mode. When Timer 2 is configured in auto-reload mode, it holds the low byte of the reload value.

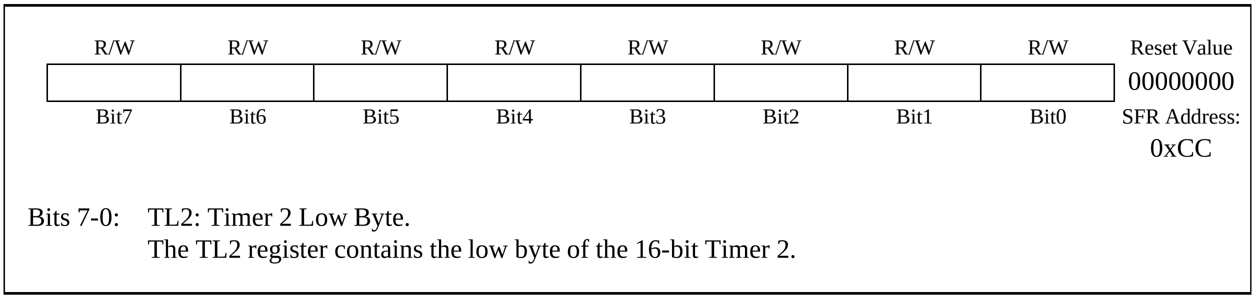
RCAP2H: Timer 2 Capture Register High Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xCB

Bits 7-0: RCAP2H: Timer 2 Capture Register High Byte.

The RCAP2H register captures the high byte of Timer 2 when Timer 2 is configured in capture mode. When Timer 2 is configured in auto-reload mode, it holds the high byte of the reload value.

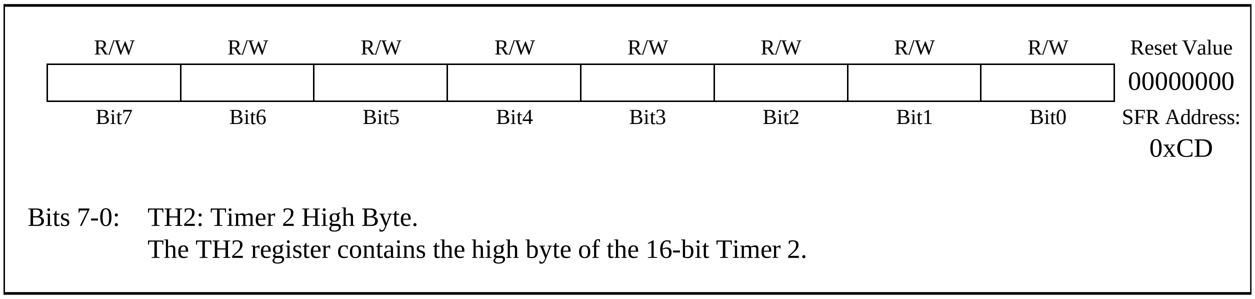
TL2: Timer 2 Low Byte



R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xCC

Bits 7-0: TL2: Timer 2 Low Byte.
The TL2 register contains the low byte of the 16-bit Timer 2.

TH2 Timer 2 High Byte



R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xCD

Bits 7-0: TH2: Timer 2 High Byte.
The TH2 register contains the high byte of the 16-bit Timer 2.

5.3 DRV8313 Triple half H Bridge Driver IC

Description:
The DRV8313 provides three individually controllable half-H-bridge drivers. The device is intended to drive a three-phase brushless-DC motor, although it can also be used to drive solenoids or other loads. Each output driver channel consists of N-channel power MOSFETs configured in a 1/2-H-bridge configuration. Each 1/2-H-bridge driver has a dedicated ground terminal, which allows independent external current sensing. An uncommitted comparator is integrated into the DRV8313, which allows for the construction of current-limit circuitry or other functions. Internal protection functions are provided for undervoltage, charge pump faults, overcurrent, short circuits, and overtemperature. Fault conditions are indicated by the nFAULT pin.

Features

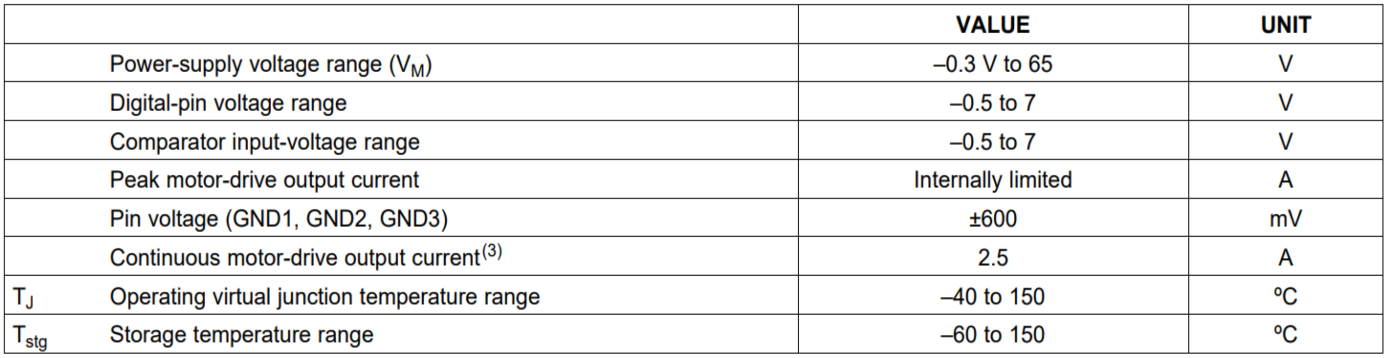
1. Triple 1/2-H Bridge Driver IC
   – 3-Phase brushless DC Motors
   – Solenoid and Brushed DC Motors
2. High Current-Drive Capability: 2.5-A Peak
3. Low MOSFET ON-Resistance
4. Independent 1/2-H-Bridge Control
5. Uncommitted Comparator Can Be Used for Current Limit or Other Functions
6. Built-In 3.3-V 10-mA LDO Regulator
7. 8-V to 60-V Operating Supply-Voltage Range
8. Sleep Mode for Standby Operation
9. Small Package and Footprint
   – 28-Pin HTSSOP (PowerPAD™ Package)
   – 36-Pin VQFN 2

Applications :

1. Camera Gimbals
2. HVAC Motors
3. Office Automation Machines
4. Factory Automation and Robotic

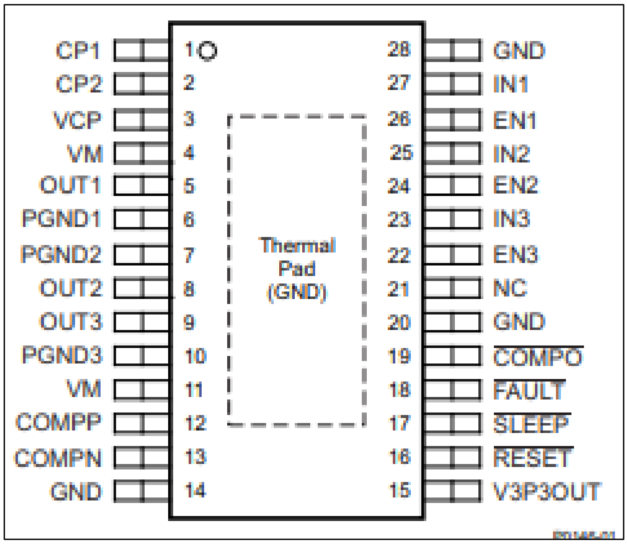
ABSOLUTE MAXIMUM RATINGS

over operating free-air temperature range (unless otherwise noted)



		VALUE	UNIT
	Power-supply voltage range (V_M)	–0.3 V to 65	V
	Digital-pin voltage range	–0.5 to 7	V
	Comparator input-voltage range	–0.5 to 7	V
	Peak motor-drive output current	Internally limited	A
	Pin voltage (GND1, GND2, GND3)	±600	mV
	Continuous motor-drive output current[3]	2.5	A
T_J	Operating virtual junction temperature range	–40 to 150	ºC
T_{stg}	Storage temperature range	–60 to 150	ºC

PWP Package(Top View):



Functional Block Diagram

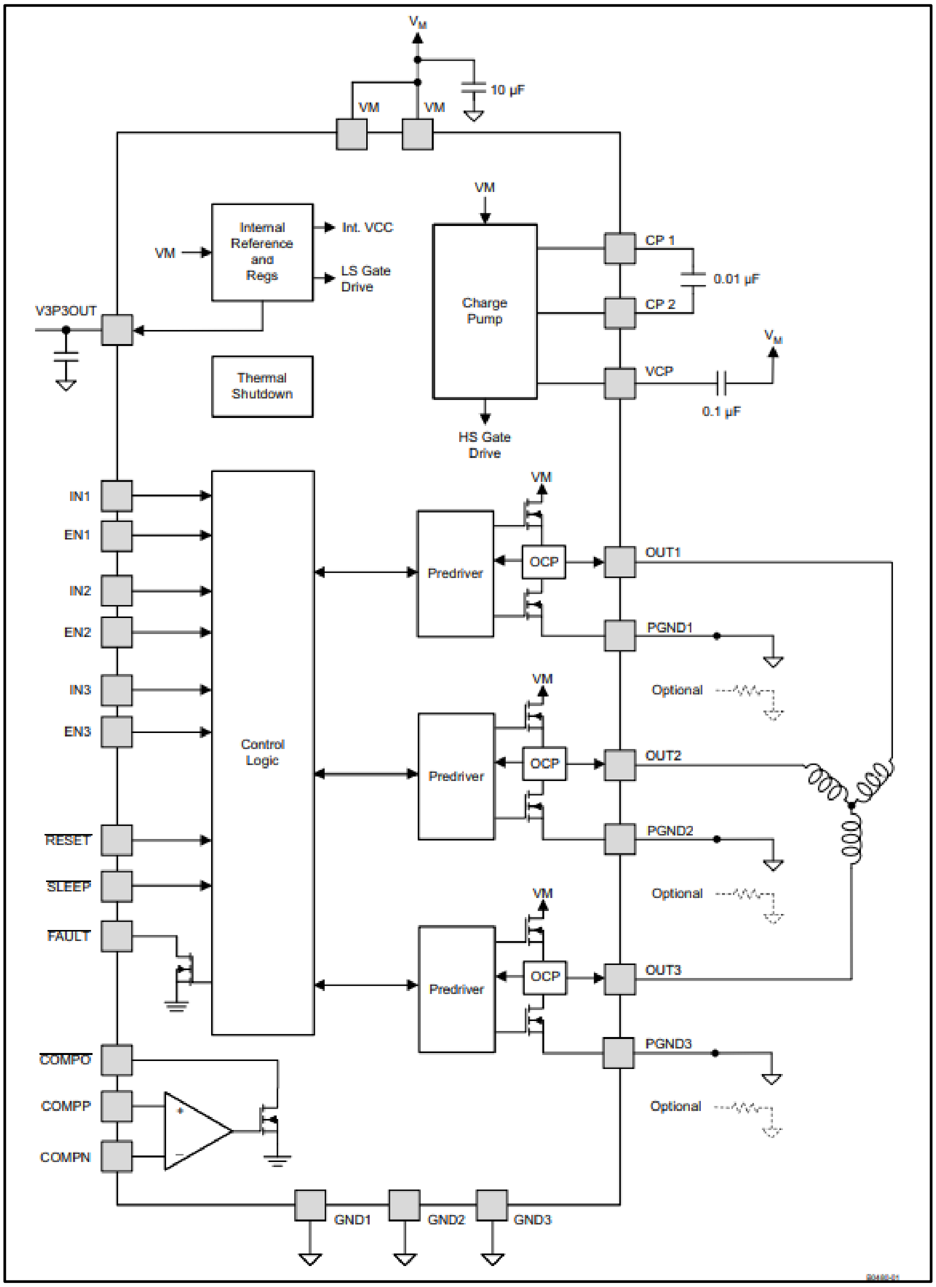
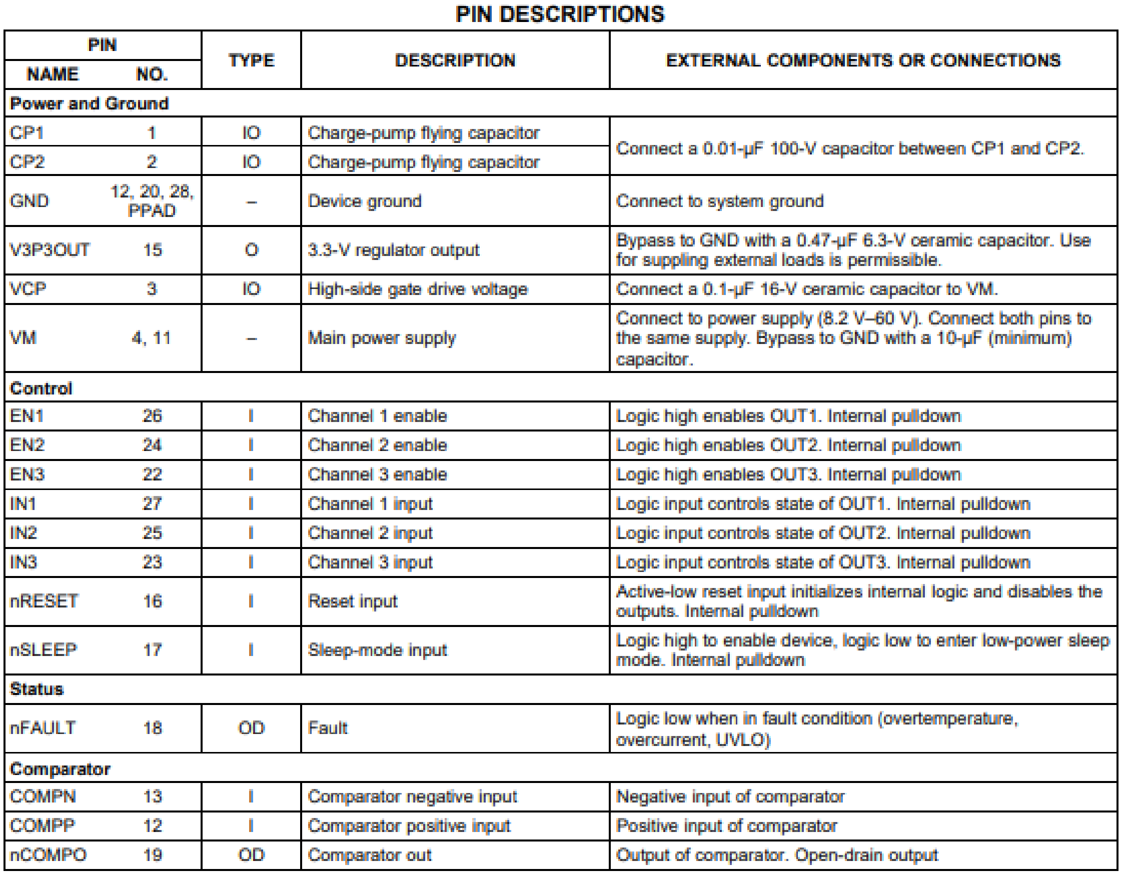


Figure : Functional Block Diagram of DRV8313



PIN DESCRIPTIONS

PIN		TYPE	DESCRIPTION	EXTERNAL COMPONENTS OR CONNECTIONS
NAME	NO.			
Power and Ground				
CP1	1	IO	Charge-pump flying capacitor	Connect a 0.01-μF 100-V capacitor between CP1 and CP2.
CP2	2	IO	Charge-pump flying capacitor	
GND	12, 20, 28, PPAD	–	Device ground	Connect to system ground
V3P3OUT	15	O	3.3-V regulator output	Bypass to GND with a 0.47-μF 6.3-V ceramic capacitor. Use for suppling external loads is permissible.
VCP	3	IO	High-side gate drive voltage	Connect a 0.1-μF 16-V ceramic capacitor to VM.
VM	4, 11	–	Main power supply	Connect to power supply (8.2 V–60 V). Connect both pins to the same supply. Bypass to GND with a 10-μF (minimum) capacitor.
Control				
EN1	26	I	Channel 1 enable	Logic high enables OUT1. Internal pulldown
EN2	24	I	Channel 2 enable	Logic high enables OUT2. Internal pulldown
EN3	22	I	Channel 3 enable	Logic high enables OUT3. Internal pulldown
IN1	27	I	Channel 1 input	Logic input controls state of OUT1. Internal pulldown
IN2	25	I	Channel 2 input	Logic input controls state of OUT2. Internal pulldown
IN3	23	I	Channel 3 input	Logic input controls state of OUT3. Internal pulldown
nRESET	16	I	Reset input	Active-low reset input initializes internal logic and disables the outputs. Internal pulldown
nSLEEP	17	I	Sleep-mode input	Logic high to enable device, logic low to enter low-power sleep mode. Internal pulldown
Status				
nFAULT	18	OD	Fault	Logic low when in fault condition (overtemperature, overcurrent, UVLO)
Comparator				
COMPN	13	I	Comparator negative input	Negative input of comparator
COMPP	12	I	Comparator positive input	Positive input of comparator
nCOMPO	19	OD	Comparator out	Output of comparator. Open-drain output

Output Configurations and Connections

The typical application for the DRV8313 is to drive a 3-phase brushless motor. In this application, the three outputs connect to the three motor leads, as shown in Figure

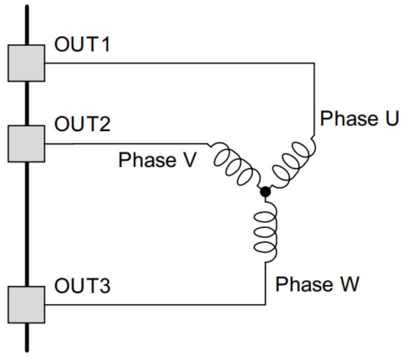
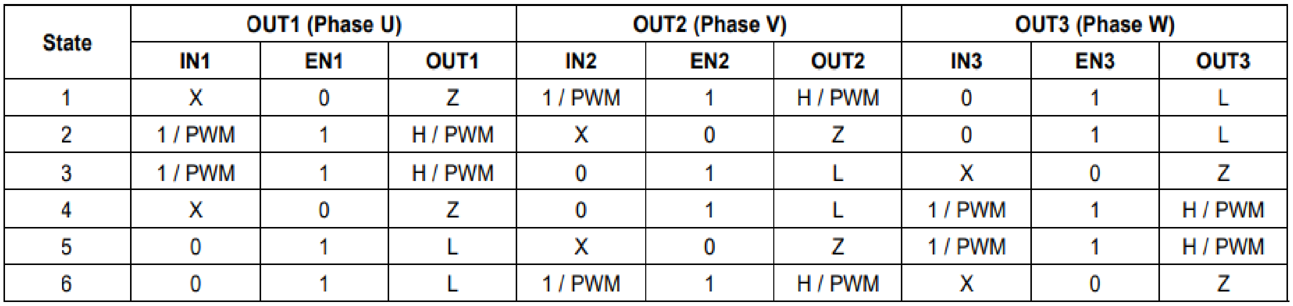


Figure : Three Phase Motor Connections

The device achieves standard 120° (also called trapezoidal or block) commutation, using synchronous rectification, by following the states shown in Table 1



State	OUT1 (Phase U)			OUT2 (Phase V)			OUT3 (Phase W)		
	IN1	EN1	OUT1	IN2	EN2	OUT2	IN3	EN3	OUT3
1	X	0	Z	1 / PWM	1	H / PWM	0	1	L
2	1 / PWM	1	H / PWM	X	0	Z	0	1	L
3	1 / PWM	1	H / PWM	0	1	L	X	0	Z
4	X	0	Z	0	1	L	1 / PWM	1	H / PWM
5	0	1	L	X	0	Z	1 / PWM	1	H / PWM
6	0	1	L	1 / PWM	1	H / PWM	X	0	Z

On can implement asynchronous rectification by also applying the PWM signal to the enable inputs.

09.Work table

Week	Date	Nature of work

01	Mon;
	01 Feb 2021;
	To
	Fri;
	05 Feb 2021;
		1. Pelletron, microcontroller, and video and pdf
		2. Numbering and coding conversion
		3. Micro-processor features, mate - rials details, Microcontroller. Block diagram, features and pin configuration
		4. RISC and CISC comparison, features, architecture.
		5. Arduino Uno details and micro controller in detail.

02	Mon;
	08 Feb 2021;
	To
	Fri;
	12 Feb 2021;
		1. Microcontroller Timer, counter and interrupts.
		2. Assembler directives, Mnemonic address
		3. Microcontroller Harvard Architecture and von Neumann Architecture, (Memory Space block DIA)
		4. Modes, TMOD Register, RAM memory Space Allocation in 8051
		5. Microcontroller 8051 I/O ports
		6. DRV8313 25A Triple 1/2 Bridge Driver (Block diagram)

03	Mon;
	15 Feb 2021;
		Microcontroller 8051 C Ports, Timer and interrupt

04	Mon;
	22 Feb 2021;
	To
	Fri;
	26 Feb 2021;
		1. DRV8313 Bridge driver, Functional Block diagram, Features, description
		2. DRV8313 PWP Package pin diagram
		3. DRV8313 Description of pins
		4. DRV8313 Absolute maximum ratings.
		5. DRV 8313 Functional Description

05	Mon;
	01 Mar 2021;
	To
	Fri;
	05 Mar 2021;
		1. Output Stage of DRV8313
		2. Analog peripherals of DRV 8313 8-bit ADC Two 12-hit DACS
		3. clock Sources High Speed 8051 uc core Digital peripherals Temperature Range
		4. TQFP - 100 pinout Diagram
		5. 12. Bit Analog to Digital programmable Digital I/O crossbar, Digital crossbar diagram

06	Mon;
	08 Mar 2021;
	To
	Fri;
	12 Mar 2021;
		1. C8051F02X. - Voltage reference
		2. System overview of Micro Controller (C8051F020)
		3. about maximum Rating of DRV8313

07	Mon;
	15 Mar 2021;
	To
	Fri;
	19 Mar 2021;
		1. Input voltage, current and frequency output of the microcontroller C805120
		2. Special function register (SFR) memory map of C8051F120
		3. Interfacing of C8091F120 and DRV8313 IC'S
		4. Lower port I/O Functional Block diagram of C8051F120
		5. crossbar of C8051F120

08	Mon;
	22 Mar 2021;
	To
	Fri;
	26 Mar 2021;
		1. Crossbar
		2. Timer location
		3. Crossbar in detail
		4. Ports & Pins
		5. Programming

09	Mon;
	29 March 2021;
	To
	Fri;
	01 April 2021;
		1. Programing crossbars and system clock.
		2. Programing ASM

10	Mon;
	05 April 2021;
	To
	Fri;
	09 April 2021;
		1. Programing.
		2. Timer programing.
		3. Timer calculations
		.

11	Mon;
	12 April 2021;
	To
	Fri;
	16 April 2021;
		1. Programing on generating 1 kHz through port P1.7.
		2. Calculations on machine cycle and counter value.

12	Mon;
	19 April 2021;
	To
	Fri;
	23 April 2021;
		1. Using interrupts, programing by generating 1 kHz at port 7,7

13	Mon;
	26 April 2021;
	To
	Fri;
	30 April 2021;
		1. Programing using interrupts.
		2. Interrupt (IE) and Interrupt priority (IP) in brief.

14	Mon;
	03 May 2021;
	To
	Fri;
	07 May 2021;
		1. 3 Phase programing.
		2. 120 Phase shift programing using Interrupts.

15	Mon;
	10 May 2021;
	To
	Fri;
	14 May 2021;
		1. 120 Phase shift program.
		2. Delay calculations for each phase signal,

16	Mon;
	17 May 2021;
	To
	Fri;
	21 May 2021;
		1. Graph making on 3 phase shift signals each of 1kHz.

17	Mon;
	24 May 2021;
	To
		1. Programing.

18	Mon;
	31 May 2021;
	To
	Fri;
	04 June 2021;
		1. Programing.
		2. DRV8313 Data sheet study.

19	Mon;
	07 June 2021;
	To
	Fri;
	11 June 2021;
		1. Viva preparation.
		2. Viva

20	Mon;
	14 June 2021;
	To
	Fri;
	18 June 2019;
		1. Report writing.
		2. Internship report writing

10. Conclusion

During our work experience at Tata Institute of Fundamental Research, we were fortunate enough to gain firsthand experience and we learned many different sides of what goes into an industry the general process of how a project is initially planned, developed and completed; as well as how much work and detail goes into every stage. Another valuable lesson we have learned during these 24 weeks, is the many different types of work an engineer has to perform which in turn, have provided us with more insight into the different types of the role and responsibilities that we could perform, as an 'Electronics Engineer'.

Recalling back to our first day at Tata Institute of Fundamental Research, we feel it was one of the most memorable days of our time at work after being introduced to everyone. They handed us a task to work on in which they had asked us for recommendation to improve certain aspects of the given system. They were very open and happy for us to contribute as much as we wanted to and were open to any suggestions and ideas. This instantly made us feel comfortable and definitely made us feel as though we were part of the team.

There were many days that we were busy where a given order with deadline had to be completed urgently and these days were stressful. This in some ways reminded us a lot of night study sessions at the college.

Our overall work experience at Tata Institute of Fundamental Research was very fruitful. We were very happy with the number of things that we have learned and the experience we gained in the 24 weeks of being a 'Student Engineer' at the industry. This In-plant Training has proved very beneficial to us.

We ended up learning a lot more than we thought. We are very happy and proud that we have completed our In-plant training.

11. References:

Text/ Reference Books:

(1) Basic Electronics & Linear Circuits (N. N. Bhargav, D. C. Kulshreshtha, S. C. Gupta). TMH Publishing, New Delhi. 2/e (© 2013)

(2) The 8051 Microcontroller and Embedded Systems (Mohammed Ali Mazidi, Janice G. Mazidi et al). Prentice Hall of India Ltd; New Delhi. 2/e (© 2005)

Datasheets:

(1) URL: https://www.silabs.com/documents/public/data-sheets/C8051F02x.pdf
SILICON LABS C8051F020/1/2/3 8K ISP FLASH MCU Family

(2) URL: Drv8313 datasheet.pdf
DRV8313 ½ - H Bridge Driver IC(Texas Instruments)