CHAPTER: 1
INTRODUCTION TO LAND MINES
1.1 ABOUT THE LANDMINES:

Landmines are basically explosive devices that are designed to explode when triggered by pressure or a tripwire. These devices are typically found on or just below the surface of the ground. The purpose of mines when used by armed forces is to disable any person or vehicle that comes into contact with it by an explosion or fragments released at high speeds.

Landmines are easy-to-make, cheap and effective weapons that can be deployed easily over large areas to prevent enemy movements. Mines are typically placed in the ground by hand, but there are also mechanical minelayers that can plow the earth and drop and bury mines at specific intervals.

Mines are often laid in groups, called mine fields, and are designed to prevent the enemy from passing through a certain area, or sometimes to force an enemy through a particular area.

1.2 TYPES OF LAND MINES:

An army also will use landmines to slow an enemy until reinforcements can arrive. While more than 350 varieties of mines exist, they can be broken into two categories:

- Anti-personnel (AP) mines
- Anti-tank (AT) mines

1) Anti-personnel Mines

Anti-personnel landmines are designed specifically to reroute or push back foot soldiers from a given geographic area. These mines can kill or disable their victims, and are activated by pressure, tripwire or remote detonation. There are also smart mines, which automatically de-activate themselves after a certain amount of time.
These are the most common types of mines currently used by the U.S. military.

Anti-personnel mines fit into three basic categories:

- (i) **Blast** - The most common type of mine, blast mines are buried no deeper than a few centimeters and are generally triggered by someone stepping on the **pressure plate**, applying about 11 to 35.3 pounds (5 to 16 kg) of pressure. These mines are designed to destroy an object in close proximity, such as a person's foot or leg. A blast mine is designed to break the targeted object into fragments.

- (ii) **Bounding** - Usually buried with only a small part of the **igniter** protruding from the ground, these mines are pressure or tripwire activated. You may also hear this type of mine referred to as a "**Bouncing Betty**." When activated, the igniter sets off a **propelling charge**, lifting the mine about 1 meter into the air. The mine then ignites a main charge, causing injury to a person's head and chest.

- (iii) **Fragmentation** - These mines release fragments in all directions, or can be arranged to send fragments in one direction (**directional fragmentation mines**). These mines can cause injury up to 200 meters away. The fragments used in the mines are either metal or glass. Fragmentation mines can be bounding or ground-based.

2) Anti-tank (AT) mines
The development of tanks during World War I led to **anti-tank mines**, and anti-personnel mines were developed to prevent enemy armies from moving anti-tank mines.

Anti-tank mines are very larger which are pressure activated, but are typically designed so that the footstep of a person won't detonate them. Most anti-tank mines require a high applied pressure of 348.33 pounds (158 kg) to 745.16 pounds (338 kg) in order to detonate used in tanks and other military vehicles. Let's take a closer look at one of these anti-tank mines.

![Anti-tank mine](image)

Anti-tank mines contain several times more explosive material than anti-personnel mines. There is enough explosive in an anti-tank mine to destroy a tank or truck, as well as kill people in or around the vehicle.
CHAPTER: 2

AT89C51 MICRO CONTROLLER

AT89C51

2.1 FEATURES:

- Compatible with MCS-51™ Products
- 4K Bytes of In-System Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
• Fully Static Operation: 0 Hz to 24 MHz
• Three-level Program Memory Lock
• 128 x 8-bit Internal RAM
• 32 Programmable I/O Lines
• Two 16-bit Timer/Counters
• Six Interrupt Sources
• Programmable Serial Channel
• Low-power Idle and Power-down Modes

DESCRIPTION:
The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel’s high-density nonvolatile memory technology and is Compatible with the industry-standard MCS-51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

BLOCK DIAGRAM:
Figure 2.1: Block Diagram of AT89C51

2.2 PIN DIGRAM:
Figure 2.2: Pin Diagram of AT89C51

PIN DESCRIPTION

VCC:
Supply voltage.

GND:
Ground.

Port 0:

Port 0 is an 8-bit open-drain bi-directional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high impedance inputs. Port 0 may also be configured to be the multiplexed low order address/data bus during accesses to external program and data memory. In this mode P0 has internal pull-ups. Port 0 also receives the code bytes during Flash programming, and outputs the code bytes during program verification. External pull-ups are required during program verification.
**Port 1:**

Port 1 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (IIL) because of the internal pullups. Port 1 also receives the low-order address bytes during Flash programming and verification.

**Port 2:**

Port 2 is an 8-bit bi-directional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (IIL) because of the internal pullups. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that uses 16-bit addresses (MOVX @ DPTR). In this application, it uses strong internal pull-ups. When emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register. Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

**Port 3:**

Port 3 is an 8-bit bi-directional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (IIL) because of the pullups. Port 3 also serves the functions of various special features of the AT89C51 as listed below:
Table 2.1 Features of AT89C51

Port 3 also receives some control signals for Flash programming and verification.

**RST:**

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

**ALE/PROG:**

Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (PROG) during Flash programming. In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory. If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

**PSEN:**

<table>
<thead>
<tr>
<th>Port Pin</th>
<th>Alternate Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>P3.0</td>
<td>RXD (serial input port)</td>
</tr>
<tr>
<td>P3.1</td>
<td>TXD (serial output port)</td>
</tr>
<tr>
<td>P3.2</td>
<td>INT0 (external interrupt 0)</td>
</tr>
<tr>
<td>P3.3</td>
<td>INT1 (external interrupt 1)</td>
</tr>
<tr>
<td>P3.4</td>
<td>T0 (timer 0 external input)</td>
</tr>
<tr>
<td>P3.5</td>
<td>T1 (timer 1 external input)</td>
</tr>
<tr>
<td>P3.6</td>
<td>WR (external data memory write strobe)</td>
</tr>
<tr>
<td>P3.7</td>
<td>RD (external data memory read strobe)</td>
</tr>
</tbody>
</table>
Program Store Enable is the read strobe to external program memory. When the AT89C51 is executing code from external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory.

**EA/VPP:**

External Access Enable. EA must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, EA will be internally latched on reset. EA should be strapped to VCC for internal program executions. This pin also receives the 12-volt programming enable voltage (VPP) during Flash programming, for parts that require 12-volt VPP.

**XTAL1:**

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

**XTAL2:**

Output from the inverting oscillator amplifier.

**Oscillator Characteristics:**

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the
external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

2.3 MODES OF OPERATION

In general the modes of operation of Micro controller are two modes. They are

- Ideal Mode
- Power down Mode

**Ideal Mode:**

In idle mode, the CPU puts itself to sleep while all the onchip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset. It should be noted that when idle is terminated by a hard ware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

**Figure 2.3. Oscillator Connections**
Note: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

### Table 2.2 Status of External pins during Idle and Power-down Modes

<table>
<thead>
<tr>
<th>Mode</th>
<th>Program Memory</th>
<th>ALE</th>
<th>PSEN</th>
<th>PORT0</th>
<th>PORT1</th>
<th>PORT2</th>
<th>PORT3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idle</td>
<td>Internal</td>
<td>1</td>
<td>1</td>
<td>Data</td>
<td>Data</td>
<td>Data</td>
<td>Data</td>
</tr>
<tr>
<td>Idle</td>
<td>External</td>
<td>1</td>
<td>1</td>
<td>Float</td>
<td>Data</td>
<td>Address</td>
<td>Data</td>
</tr>
<tr>
<td>Power-down</td>
<td>Internal</td>
<td>0</td>
<td>0</td>
<td>Data</td>
<td>Data</td>
<td>Data</td>
<td>Data</td>
</tr>
<tr>
<td>Power-down</td>
<td>External</td>
<td>0</td>
<td>0</td>
<td>Float</td>
<td>Data</td>
<td>Data</td>
<td>Data</td>
</tr>
</tbody>
</table>

### Figure 2.4: External Clock Drive Configuration

**Power-down Mode:**
In the power-down mode, the oscillator is stopped, and the instruction that invokes power-down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power-down mode is terminated. The only exit from power-down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before VCC is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

**Programming the Flash:**

The AT89C51 is normally shipped with the on-chip Flash memory array in the erased state (that is, contents = FFH) and ready to be programmed. The programming interface accepts either a high-voltage (12-volt) or a low-voltage (VCC) program enable signal. The low-voltage programming mode provides a convenient way to program the AT89C51 inside the user’s system, while the high-voltage programming mode is compatible with conventional third party Flash or EPROM programmers. The AT89C51 is shipped with either the high-voltage or Low-voltage programming mode enabled. The respective top-side marking and device signature codes are listed in the following table.

<table>
<thead>
<tr>
<th>Top-side Mark</th>
<th>( V_{pp} = 12V )</th>
<th>( V_{pp} = 5V )</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT89C51 xxxx yyy yyww</td>
<td>AT89C51 xxxx-5 yyww</td>
<td></td>
</tr>
<tr>
<td>Signature</td>
<td>(030H) = 1EH (031H) = 51H (032H) = FFH</td>
<td>(030H) = 1EH (031H) = 51H (032H) = 05H</td>
</tr>
</tbody>
</table>

**Table 2.3 Top side marking and device signature codes**
The AT89C51 code memory array is programmed byte-by-byte in either programming mode. To program any nonblank byte in the on-chip Flash Memory, the entire memory must be erased using the Chip Erase Mode.

2.4 PROGRAMMING ALGORITHM:

Before programming the AT89C51, the address, data and control signals should be set up according to the Flash programming mode table and Figure 3 and Figure 4. To program the AT89C51, take the following steps.

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise EA/VPP to 12V for the high-voltage programming mode.
5. Pulse ALE/PROG once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 1.5-4 ms. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

Data Polling:

The AT89C51 features Data Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written datum on PO.7. Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. Data Polling may begin any time after a write cycle has been initiated.

Ready/Busy:
The progress of byte programming can also be monitored by the RDY/BSY output signal. P3.4 is pulled low after ALE goes high during programming to indicate BUSY. P3.4 is pulled high again when programming is done to indicate READY.

**Program Verify:**

If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

**Chip Erase:**

The entire Flash array is erased electrically by using the proper combination of control signals and by holding ALE/PROG low for 10 ms. The code array is written with all “1”s. The chip erase operation must be executed before the code memory can be re-programmed.

**2.5 PROGRAMMING INTERFACE:**

Every code byte in the Flash array can be written and the entire array can be erased by using the appropriate combination of control signals. The write operation cycle is self timed and once initiated, will automatically time itself to completion. All major programming vendors offer worldwide support for the Atmel microcontroller series.
Table 2.4 Flash programming modes

The programming and verify sections for a microcontroller is must to operate in required manner.

Therefore the pin connections during these sections are shown below.

Figure 2.5: Pin Connections
The timing waveforms of both Flash programming and verification at Low voltage mode (V_{PP}=5V) is as shown in the below figure.

**Flash Programming and Verification Waveforms - Low-voltage Mode (V_{PP} = 5V)**

---

**2.6 TIMER/COUNTERS:**

The Atmel 89C51 Microcontroller implements two general purpose, 16-bit timers/ counters. They are identified as Timer 0 and Timer 1, and can be independently configured to operate in a variety of modes as a timer or as an event counter. When operating as a timer, the timer/counter runs for a programmed length of time, and then issues an interrupt request. When operating as a counter, the timer/counter counts negative transitions on an external pin. After a preset number of counts, the counter issues an interrupt request.

The various operating modes of each timer/counter are described in the following sections.

**Timer/Counter Operations:**
A basic operation consists of timer registers THx and TLx (x= 0, 1) connected in cascade to form a 16-bit timer. Setting the run control bit (TRx) in TCON register turns the timer on by allowing the selected input to increment TLx. When TLx overflows it increments THx; when THx overflows it sets the timer overflow flag (TFx) in TCON register. Setting the TRx does not clear the THx and TLx timer registers. Timer registers can be accessed to obtain the current count or to enter preset values. They can be read at any time but TRx bit must be cleared to preset their values, otherwise the behavior of the timer/counter is unpredictable.

The C/Tx# control bit (in TCON register) selects timer operation, or counter operation, by selecting the divided-down peripheral clock or external pin Tx as the source for the counted signal. TRx bit must be cleared when changing the mode of operation; otherwise the behavior of the timer/counter is unpredictable. For timer operation (C/Tx# = 0), the timer register counts the divided-down peripheral clock. The timer register is incremented once every peripheral cycle (6 peripheral clock periods). The timer clock rate is FPER / 6, i.e. FOSC / 12 in standard mode or FOSC / 6 in X2 mode. For counter operation (C/Tx# = 1), the timer register counts the negative transitions on the Tx external input pin. The external input is sampled every peripheral cycle. When the sample is high in one cycle and low in the next one, the counter is incremented. Since it takes 2 cycles (12 peripheral clock periods) to recognize a negative transition, the maximum count rate is FPER / 12, i.e. FOSC / 24 in standard mode or FOSC / 12 in X2 mode. There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it should be held for at least one full peripheral cycle.

In addition to the “timer” or “counter” selection, Timer 0 and Timer 1 have four operating modes from which to select which are selected by bit-pairs (M1, M0) in TMOD. Modes 0, 1, and 2 are the same for both timer/counters. Mode 3 is different. The four operating modes are described below.

2.6.1 Timer 0
Timer 0 functions as either a timer or event counter in four modes of operation. Figure 9 to Figure 12 show the logical configuration of each mode. Timer 0 is controlled by the four lower bits of the TMOD register and bits 0, 1, 4 and 5 of the TCON register. TMOD register selects the method of timer gating (GATE0), timer or counter operation (T/C0#) and mode of operation (M10 and M00). The TCON register provides timer 0 control functions: overflow flag (TF0), run control bit (TR0), interrupt flag (IE0) and interrupt type control bit (IT0). For normal timer operation (GATE0= 0), setting TR0 allows TL0 to be incremented by the selected input. Setting GATE0 and TR0 allows external pin INT0# to control timer operation. Timer 0 overflow (count rolls over from all 1s to all 0s) sets TF0 flag, generating an interrupt request. It is important to stop timer/counter before changing mode.

Mode 0 (13-bit Timer)

Mode 0 configures timer 0 as a 13-bit timer which is set up as an 8-bit timer (TH0 register) with a modulo 32 pre-scaler implemented with the lower five bits of the TL0 register. The upper three bits of TL0 register are indeterminate and should be ignored. Pre-scaler overflow increments the TH0 register. As the count rolls over from all 1’s to all 0’s, it sets the timer interrupt flag TF0. The counted input is enabled to the Timer when TR0 = 1 and either GATE = 0 or INT0 = 1. (Setting GATE = 1 allows the Timer to be controlled by external input INT0, to facilitate pulse width measurements). TR0 is a control bit in the Special Function register TCON. GATE is in TMOD.

The 13-bit register consists of all 8 bits of TH0 and the lower 5 bits of TL0. The upper 3 bits of TL0 are indeterminate and should be ignored. Setting the run flag (TR0) does not clear the registers. Mode 0 operation is the same for Timer 0 as for Timer 1. Substitute TR0, TF0 and INT0 for the corresponding Timer 1 signals in Table 10.
There are two different GATE bits, one for Timer 1 (TMOD.7) and one for Timer 0 (TMOD.3).

Figure 2.6: Timer/Counter 0 in Mode 0

**Mode 1 (16-bit Timer)**

Mode 1 is the same as Mode 0, except that the Timer register is being run with all 16 bits. Mode 1 configures timer 0 as a 16-bit timer with the TH0 and TL0 registers connected in cascade. The selected input increments the TL0 register.

Figure 2.7: Timer/Counter 0 in Mode 1

**Mode 2 (8-bit Timer with Auto-Reload)**
Mode 2 configures timer 0 as an 8-bit timer (TL0 register) that automatically reloads from the TH0 register. TL0 overflow sets TF0 flag in the TCON register and reloads TL0 with the contents of TH0, which is preset by software. When the interrupt request is serviced, hardware clears TF0. The reload leaves TH0 unchanged. The next reload value may be changed at any time by writing it to the TH0 register. Mode 2 operation is the same for Timer/Counter 1.

Figure 2.8: Timer/Counter 0 in Mode 2

Mode 3 (Two 8-bit Timers)

Mode 3 configures timer 0 so that registers TL0 and TH0 operate as separate 8-bit timers. This mode is provided for applications requiring an additional 8-bit timer or counter. TL0 uses the timer 0 control bits C/T0# and GATE0 in the TMOD register, and TR0 and TF0 in the TCON register in the normal manner. TH0 is locked into a timer function (counting FPER /6) and takes over use of the timer 1 interrupt (TF1) and run control (TR1) bits. Thus, operation of timer 1 is restricted when timer 0 is in mode 3.
2.6.2 Timer 1

Timer 1 is identical to timer 0, except for mode 3, which is a hold-count mode. The following comments help to understand the differences:

1. Timer 1 functions as either a timer or event counter in three modes of operation. Figure 9 to Figure 11 show the logical configuration for modes 0, 1, and 2. Timer 1’s mode 3 is a hold-count mode.

2. Timer 1 is controlled by the four high-order bits of the TMOD register and bits 2, 3, 6 and 7 of the TCON register. The TMOD register selects the method of timer gating (GATE1), timer or counter operation (C/T1#) and mode of operation (M11 and M01). The TCON register provides timer 1 control functions: overflow flag (TF1), run control bit (TR1), interrupt flag (IE1) and interrupt type control bit (IT1).

3. Timer 1 can serve as the baud rate generator for the serial port. Mode 2 is best suited for this purpose.

4. For normal timer operation (GATE1 = 0), setting TR1 allows TL1 to be incremented by the selected input. Setting GATE1 and TR1 allows external pin INT1# to control timer operation.

5. Timer 1 overflows (count rolls over from all 1s to all 0s) sets the TF1 flag generating an interrupt request.
6. When timer 0 is in mode 3, it uses timer 1’s overflow flag (TF1) and run control bit (TR1). For this situation, use timer 1 only for applications that do not require an interrupt (such as a baud rate generator for the serial port) and switch timer 1 in and out of mode 3 to turn it off and on.

7. It is important to stop timer/counter before changing modes.

**Mode 0 (13-bit Timer)**

Mode 0 configures Timer 1 as a 13-bit timer, which is set up as an 8-bit timer (TH1 register) with a modulo-32 pre-scalar implemented with the lower 5 bits of the TL1 register. The upper 3 bits of the TL1 register are ignored. Pre-scalar overflow increments the TH1 register.

**Mode 1 (16-bit Timer)**

Mode 1 configures Timer 1 as a 16-bit timer with the TH1 and TL1 registers connected in cascade. The selected input increments the TL1 register.

![Figure 2.10: Timer/Counter 1 in Mode 1](image)

Figure 2.10: Timer/Counter 1 in Mode 1
**Mode 2 (8-bit Timer with Auto Reload)**

Mode 2 configures Timer 1 as an 8-bit timer (TL1 register) with automatic reload from the TH1 register on overflow. TL1 overflow sets the TF1 flag in the TCON register and reloads TL1 with the contents of TH1, which is preset by software. The reload leaves TH1 unchanged.

![Figure 2.11: Timer/Counter 1 in Mode 2](image)

**Mode 3 (Halt)**

Placing Timer 1 in mode 3 causes it to halt and hold its count. This can be used to halt Timer 1 when TR1 run control bit is not available i.e., when Timer 0 is in mode 3.

**Interrupt**

Each timer handles one interrupt source; that is the timer overflow flag TF0 or TF1. This flag is set every time an overflow occurs. Flags are cleared when vectoring to the timer interrupt routine. Interrupts are enabled by setting ETx bit in IE0 register. This assumes interrupts are globally enabled by setting EA bit in the IE0 register.
Figure 2.12 Interrupt source circuit

<table>
<thead>
<tr>
<th>Bit Number</th>
<th>Bit Mnemonic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>TF1</td>
<td>Timer 1 Overflow Flag&lt;br&gt;Cleared by hardware when processor vectors to interrupt routine. Set by hardware on timer/counter overflow, when the timer 1 register overflows.</td>
</tr>
<tr>
<td>6</td>
<td>TR1</td>
<td>Timer 1 Run Control Bit&lt;br&gt;Clear to turn off timer/counter 1. Set to turn on timer/counter 1.</td>
</tr>
<tr>
<td>5</td>
<td>TF0</td>
<td>Timer 0 Overflow Flag&lt;br&gt;Cleared by hardware when processor vectors to interrupt routine. Set by hardware on timer/counter overflow, when the timer 0 register overflows.</td>
</tr>
<tr>
<td>4</td>
<td>TR0</td>
<td>Timer 0 Run Control Bit&lt;br&gt;Clear to turn off timer/counter 0. Set to turn on timer/counter 0.</td>
</tr>
<tr>
<td>3</td>
<td>IE1</td>
<td>Interrupt 1 Edge Flag&lt;br&gt;Cleared by hardware when interrupt is processed if edge-triggered (see IT1). Set by hardware when external interrupt is detected on INT1# pin.</td>
</tr>
<tr>
<td>2</td>
<td>IT1</td>
<td>Interrupt 1 Type Control Bit&lt;br&gt;Clear to select low level active (level triggered) for external interrupt 1 (INT1#). Set to select falling edge active (edge triggered) for external interrupt 1.</td>
</tr>
<tr>
<td>1</td>
<td>IE0</td>
<td>Interrupt 0 Edge Flag&lt;br&gt;Cleared by hardware when interrupt is processed if edge-triggered (see IT0). Set by hardware when external interrupt is detected on INT0# pin.</td>
</tr>
<tr>
<td>0</td>
<td>IT0</td>
<td>Interrupt 0 Type Control Bit&lt;br&gt;Clear to select low level active (level triggered) for external interrupt 0 (INT0#). Set to select falling edge active (edge triggered) for external interrupt 0.</td>
</tr>
</tbody>
</table>

Table 2.5: TCON Register – TCON(S:88h)
### Table 2.6: TMOD - Timer/Counter 0 and 1 Modes

<table>
<thead>
<tr>
<th>Bit Number</th>
<th>Bit Mnemonic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>GATE1</td>
<td>Timer 1 Gating Control Bit - Clear to enable timer 1 whenever the TR1 bit is set. Set to enable timer 1 only while the INT1# pin is high and TR1 bit is set.</td>
</tr>
<tr>
<td>6</td>
<td>C/T1#</td>
<td>Timer 1 Counter/Timer Select Bit - Clear for timer operation: timer 1 counts the divided-down system clock. Set for counter operation: timer 1 counts negative transitions on external pin T1.</td>
</tr>
<tr>
<td>5</td>
<td>M11</td>
<td>Timer 1 Mode Select Bits - Operating mode M11 M10</td>
</tr>
<tr>
<td>4</td>
<td>M0X</td>
<td>Mode 0: 8-bit timer/counter (TH1) with 5-bit prescaler (TL1). Mode 1: 16-bit timer/counter. Mode 2: 8-bit auto-reload timer/counter (TL1), Reloaded from TH1 at overflow. Mode 3: Timer 1 halted. Retains count.</td>
</tr>
<tr>
<td>3</td>
<td>GATE0</td>
<td>Timer 0 Gating Control Bit - Clear to enable timer 0 whenever the TR0 bit is set. Set to enable timer counter 0 only while the INT0# pin is high and the T0F bit is set.</td>
</tr>
<tr>
<td>2</td>
<td>C/T0#</td>
<td>Timer 0 Counter/Timer Select Bit - Clear for timer operation: timer 0 counts the divided-down system clock. Set for counter operation: timer 0 counts negative transitions on external pin T0.</td>
</tr>
<tr>
<td>1</td>
<td>M10</td>
<td>Timer 0 Mode Select Bit - Operating mode M10 M0X</td>
</tr>
<tr>
<td>0</td>
<td>M0X</td>
<td>Mode 0: 8-bit timer/counter (TH0) with 5-bit prescaler (TL0). Mode 1: 16-bit timer/counter. Mode 2: 8-bit auto-reload timer/counter (TL0), Reloaded from TH0 at overflow. Mode 3: Timer 2 is an 8-bit timer using timer 1's TH0 and T0F bits.</td>
</tr>
</tbody>
</table>

Reset Value = 0000 0000b

### Table 2.7: TH0 Register - TH0

Timer 0 High Byte Register

<table>
<thead>
<tr>
<th>Bit Number</th>
<th>Bit Mnemonic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0</td>
<td></td>
<td>High Byte of Timer 0.</td>
</tr>
</tbody>
</table>

Reset Value = 0000 0000b
When Timer 0 is in Mode 3, Timer 1 can be turned on and off by switching it out of and in to its own Mode 3, or can still be used by the serial port as a baud rate generator, or in fact, in any application not requiring an interrupt.
CHAPTER: 3

MODULES OF LAND MINE SWEeper ROBO

3.1 RF MODULE
In order to control the movement of robo we are using the 4 channel RF module accordingly the motions of the robo will be controlled. The 4 channel RF module consists of the transmitter and the receiver section. From the receiver four outputs are produced and these are given to the micro controller. The four channels RF remote is shown in the figure.

### 3.1.1 TRANSMITTER SECTION:

**TWS-434**: The transmitter output is up to 8mW at 433.92MHz with a range of approximately 400 foot (open area) outdoors. Indoors, the range is approximately 200 foot, and will go through most walls.....

![TWS-434A](image)

**Figure 3.1:** TWS-434 Pin Diagram

The TWS-434 transmitter accepts both linear and digital inputs, can operate from 1.5 to 12 Volts-DC, and makes building a miniature hand-held RF transmitter very easy. The TWS-434 is approximately the size of a standard postage stamp.
a. **ENCODER:**

PT2262 is a remote control encoder paired with PT2272 utilizing CMOS Technology. It encodes data and address pins into a serial coded waveform suitable for RF or IR modulation. PT2262 has a maximum of 12 bits of tri-state address pins providing up to 531,441 (or 312) address codes; thereby, drastically reducing any code collision and unauthorized code scanning possibilities.

b. **FEATURES:**

- CMOS Technology
- Low Power Consumption
- Very High Noise Immunity
- Up to 12 Tri-State Code Address Pins
• Up to 6 Data Pins
• Wide Range of Operating Voltage: Vcc= 4~15V
• Single Resistor Oscillator
• Latch or Momentary Output Type
• Available in DIP and SOP

**Block diagram:**

---

**Figure 3.3: Block Diagram of Transmission Section**

PT2262 encodes the code address and data set at A0 ~ A5 and A6/D5 ~ A11/D0 into a special waveform and outputs it to the DOUT when /TE is pulled to “0” (Low State). This waveform is fed to either the RF modulator or the IR transmitter for transmission. The transmitted radio frequency or infrared ray is received by the RF demodulator or IR receiver and reshaped to the special waveform. PT2272 is then used to decode the waveform and set the corresponding output pin(s). Thus completing a remote control encoding and decoding function.

c. **PIN DIAGRAM:**
Figure 3.4: Pin Diagram of PT2262

PIN DESCRIPTION:

<table>
<thead>
<tr>
<th>Pin Name</th>
<th>I/O</th>
<th>Description</th>
<th>Pin No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0 ~ A5</td>
<td>I</td>
<td>Code Address Pin Nos. 0 ~ 5. These six tri-state pins are detected by PT2262 to determine the encoded waveform bit 0 ~ bit 5. Each pin can be set to &quot;0&quot;, &quot;1&quot;, or &quot;F&quot; (floating).</td>
<td>1 ~ 6</td>
</tr>
<tr>
<td>A6/D5 ~ A11/D0</td>
<td>I</td>
<td>Code Address Pin Nos. 5 ~ 11/Data Pin Nos. 5 ~ 0. These six tri-state pins are detected by PT2262 to determine the encoded waveform bit 6 ~ bit 11. When these pins are used as address pins, they can be set to &quot;0&quot;, &quot;1&quot;, or &quot;F&quot; (floating). When these pins are used as data pins, they can be set only to &quot;0&quot; or &quot;1&quot;.</td>
<td>7 ~ 8</td>
</tr>
<tr>
<td>/TE</td>
<td>I</td>
<td>Transmission Enable. Active Low Signal. PT2262 outputs the encoded waveform to DOUT when this pin is pulled low.</td>
<td>14</td>
</tr>
<tr>
<td>OSC 1</td>
<td>O</td>
<td>Oscillator Pin No.1. A resistor connected between these two pins determine the fundamental frequency of the PT2262.</td>
<td>15</td>
</tr>
<tr>
<td>OSC 2</td>
<td>I</td>
<td>Oscillator Pin No.2.</td>
<td>16</td>
</tr>
<tr>
<td>DOUT</td>
<td>O</td>
<td>Data Output Pin. The encoded waveform is serially outputted to this pin. When PT2262 is not transmitting, DOUT outputs low (Vss) voltage.</td>
<td>17</td>
</tr>
<tr>
<td>Vcc</td>
<td>-</td>
<td>Positive Power Supply</td>
<td>18</td>
</tr>
<tr>
<td>Vss</td>
<td>-</td>
<td>Negative Power Supply</td>
<td>9</td>
</tr>
</tbody>
</table>

Table 3.1 Pin description of PT2262
d. ENCODING OF CODE BITS IN RF TRANSMITTER

CODE BITS:

A Code Bit is the basic component of the encoded waveform, and can be classified as either an AD (Address/Data) Bit or a SYNC (Synchronous) Bit.

Address/Data (AD) Bit Waveform. An AD Bit can be designated as Bit “0”, “1” or “f” if it is in low, high or floating state respectively. One bit waveform consists of 2 pulse cycles. Each pulse cycle has 16 oscillating time periods. For further details, please refer to the diagram below:

Synchronous (Sync.) Bit Waveform:

The Synchronous Bit Waveform is 4 bits long with 1/8 bit width pulse. Please refer to the diagram below:
**CODE WORD:**

A group of Code Bits is called a Code Word. A Code Word consists of 12 AD bits followed by one Sync Bit. The 12 AD bits are determined by the corresponding states of A0 ~ A5 and A6/D5 ~ A11/D0 pins at the time of transmission. When Data Type of PT2262 is used, the address bits will decrease accordingly.

For example: In the 3 Data Type where the address has nine (9) bits, the transmitting format is:

- **9 Address Bits**
- **3 Data Bits**
- **Sync. bit**

PT2262 / PT2272 have a maximum of twelve (12) Address Bits including the six (6) Address/Data bits. The following diagram shows the code bits with their corresponding pins.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

One Complete Code Word

0 Data: A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 Sync Bit

1 Data: A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 D0 Sync Bit

2 Data: A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 D1 D0 Sync Bit

3 Data: A0 A1 A2 A3 A4 A5 A6 A7 A8 D2 D1 D0 Sync Bit

4 Data: A0 A1 A2 A3 A4 A5 A6 A7 D3 D2 D1 D0 Sync Bit

5 Data: A0 A1 A2 A3 A4 A5 D4 D3 D2 D1 D0 Sync Bit

6 Data: A0 A1 A2 A3 A4 A5 D5 D4 D3 D2 D1 D0 Sync Bit

The Code Bits A0 ~ A5 and A6/D5 ~ A11/D0 are determined by the states of A0 ~ A5 and A6/D5 ~ A11/D0 pins. For example, when the A0 (Pin No. 1) is set to “1” (Vcc), the Code Bit A0 is synthesized as “1” bit. In the same manner, when it (A0 Pin) is set to “0” (Vss) or left floating, the Code Bit A0 is synthesized as a “0” or “f” bit respectively.

**CODE FRAME:**
A Code Frame consists of four (4) continuous Code Words. When PT2262 detects “0” on the /TE (meaning, the /TE is active “low”), it outputs a Code Frame at DOUT. If /TE is still active at the time the Code Frame transmission ends, PT2262 outputs another Code Frame. It should be noted that the Code Frame is synthesized at the time of transmission.

**SIGNAL RESISTOR OSCILLATOR:**

The built-in oscillator circuitry of PT2262 allows a precision oscillator to be constructed by connecting an external resistor between OSC1 and OSC2 pins. For PT2272 to decode correctly the received waveform, the oscillator frequency of PT2272 must be 2.5~8 times that of transmitting PT2262. The typical oscillator frequency with various resistor values for both PT2262 and PT2272.

<table>
<thead>
<tr>
<th>Suggested oscillator resistor values are shown below.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PT2262</strong></td>
</tr>
<tr>
<td>4.7 M</td>
</tr>
<tr>
<td>3.3 M</td>
</tr>
<tr>
<td>1.2 M</td>
</tr>
</tbody>
</table>

Note:

* -- Operates when PT2272's Vcc=5V to 15V
** -- Operates when PT2272's Vcc=3V to 15V

This means that if PT2272 supply voltage is lower than 5V, you need to use a lower oscillator resistor value for both PT2262 and PT2272.

**OPERATION FLOW CHART:**

1. When Power is turned on, PT2262 activates the Stand-By Mode.
2. It then checks if TE is in to “0” (Low State). If it is not in low state it remains in the Stand-By Mode. Otherwise, it transmits 4 code address and data set at A0 ~ A5 and A6/D5 ~ A11/D0 into a special waveform and outputs it to the DOUT.

3. Again it checks still if TE is in to “0” (Low State) or not. If it is in low state again it will transmit 4 word data and address if not it goes Stand-By Mode.

**Specifications:**
Table 3.2 Absolute maximum rating & DC electrical characteristics

---

e. **TRANSMISSION CIRCUIT:**
When any of A-D buttons is pressed the /TE pin is pulled low, and power is applied to both the encoder chip and transmitter module, the encoder then starts scanning and transmitting the status of the 12bits address and data serially.

3.1.2 ANTENNA CONFIGURATION:
a. WHIP ANTENNA:

The simplest antenna is the “Whip” antenna. These antennas are commonly used in applications where range is important and are also very easy to design and tune. Whip antenna is a quarter wavelength straight wire or rod (Fig. 3.6) connected directly to the Antenna pin of RX/TX. The length of a resonant quarter wavelength whip antenna may be calculated from the following formula:

\[ L \text{ (cm)} = \frac{7500}{\text{Freq. (MHz)}} \]

At 433.92 MHz, one quarter of wavelength is 17 cm.
This formula is only a starting point since the length may be shorter if the
whip is overly thick or wide, or have any kind of coating. It may need to be longer if
the ground plane is too small. These antennas are easy to tune, simply by slight
changes in length. If the antenna is installed remotely from the receiver/Transmitter
module, a 50Ω coaxial cable can be used (Fig.3.7).

A helical antenna is a wire coil usually wound from steel, copper, or brass (Fig.3.8).

Because a helical has a high Q factor, its bandwidth is very narrow and the spacing of
the coils has a pronounced effect on antenna performance. The number of turns on the coil will depend on wire size, coil diameter, and turn spacing. The numbers of turns can be determined empirically by taking a excessively long coil and tuning it by clipping until it is resonant at the desired frequency. The coil can be fine tuned by spreading or compressing the length of the coil. For 433.92 MHz use 17 turns of 1.0 mm enameled copper wire close wound on 5.0 mm diameter former, \( L = 30 \text{ mm} \) The big problem with this antenna is that it can be easily detuned by nearby objects, including a hand, so it may not be good for handheld use.

3.1.3 RECEIVER SECTION:

RWS-434: The receiver also operates at 433.92MHz, and has a sensitivity of 3\(\mu\)V. The RWS-434 receiver operates from 4.5 to 5.5 volts-DC, and has both linear and digital outputs.

![Figure 3.9: RWS434 Pin Diagram](image)

Note: For maximum range, the recommended antenna should be approximately 35cm long. To convert from centimeters to inches -- multiply by 0.3937. For 35cm, the length in inches will be approximately 35cm \( \times 0.3937 = 13.7795 \) inches long. We
tested these modules using a 14", solid, 24 gauge hobby type wire, and reached a range of over 400 foot. Your results may vary depending on your surroundings.

The circuit diagram for the receiver is shown in figure . The decoder receives serial addresses and data from the encoder that are transmitted by the RF transmitter module. It compares the serial input data three times continuously with its local addresses. If no error or unmatched codes are found, the input data codes are then decoded and transferred to the output pins D8~D11. The VT pin also goes high to indicate a valid transmission, which will turn on the LED1. The addresses of the decoder (set by S1) have to be matched with the transmitter encoder.

![Circuit Diagram](image)

**Figure 3.10:** Sample Receiver Application Circuit

**a. DECODER:**

43
PT2272 is a remote control decoder paired with PT2262 utilizing CMOS Technology. It has 12 bits of tri-state address pins providing a maximum of 531,441 (or 312) address codes; thereby, drastically reducing any code collision and unauthorized code scanning possibilities.

**Block diagram:**

![Block Diagram of Receiver Section](image)

*Figure 3.11: Block Diagram of Receiver Section*

PT2272 decodes the waveform received and fed into the DIN pin. The waveform is decoded into code word that contains the address, data and sync bits. The decoded address bits are compared with the address set at the address input pins. If both addresses match for 2 consecutive code words, PT2272 drives - (1) the data output pin(s) whose corresponding data bit(s) is then decoded to be a “1” bit, and (2) the VT output -- to high voltage (high state).
b. PIN DIAGRAM:

![Figure 3.12: Pin Diagram of PT2272](image)

**Table 3.3 Pin description of PT2272**

<table>
<thead>
<tr>
<th>Pin Name</th>
<th>I/O</th>
<th>Description</th>
<th>Pin No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0 – A5</td>
<td>I</td>
<td>Code Address Pin Nos. 0 – 5. These six tri-state pins are detected by PT2272 to determine the encoded waveform bit 0 – bit 5. Each pin can be set to &quot;0&quot;, &quot;1&quot;, or &quot;F&quot; (floating).</td>
<td>1 – 6</td>
</tr>
<tr>
<td></td>
<td>I/O</td>
<td>Code Address Pin Nos. 6 – 11</td>
<td>1 – 6</td>
</tr>
<tr>
<td>A6/D8 – A11/D0</td>
<td>I/O</td>
<td>These six pins are used as higher address input bits or data output pins depending on the version (type) of PT2272 used. When used as address inputs, these pins are tri-state input pins and each pin can be set to &quot;0&quot;, &quot;1&quot;, or &quot;F&quot; (floating). When used as output pins, these pins are driven to VCC if (1) the address decoded from the waveform that was received matches the address setting at the address input pins, and (2) the corresponding data bits received is a &quot;1&quot; bit. Otherwise, they are driven to VSS.</td>
<td>7 – 8</td>
</tr>
<tr>
<td>DIN</td>
<td>I</td>
<td>Data Input Pin. The encoded waveform received is serially fed to PT2272 at this pin.</td>
<td>14 – 15</td>
</tr>
<tr>
<td>OSC 1</td>
<td>I</td>
<td>Oscillator Pin No.1. A resistor connected between these two pins determine the fundamental frequency of PT2272.</td>
<td>15 – 16</td>
</tr>
<tr>
<td>OSC 2</td>
<td>O</td>
<td>Oscillator Pin No. 2. Valid Transmission Active High Signal. VT in high state signifies that PT2272 receives valid transmission waveform.</td>
<td>16 – 17</td>
</tr>
<tr>
<td>VT</td>
<td>O</td>
<td>Positive Power Supply.</td>
<td>18 – 19</td>
</tr>
<tr>
<td>VCC</td>
<td>-</td>
<td>Negative Power Supply.</td>
<td>9 – 9</td>
</tr>
<tr>
<td>NC</td>
<td>-</td>
<td>No connection.</td>
<td>10 – 11</td>
</tr>
</tbody>
</table>


c. **DECODING OF CODE BITS IN RF RECEIVER**
When PT2272 receives a transmission code word, it initially checks whether this is a valid transmission. For a transmission to be valid, (1) it must be a Complete Code Word, and (2) the Address Bits must match the Address Setting at the Address Pins. After two consecutive valid transmissions, PT2272 (1) drives the data pins according to the data bits received, and (2) raises VT to high voltage (high state).
LATCH OR MOMENTARY DATA OUTPUT TYPE:

PT2272 uses either the latch or the momentary data output type depending on the PT2272 version used. The latch type (PT2272-Lx) activates the data out during transmission and this data is sustained in the memory until another data is inputted or entered. A momentary type (PT2272-Mx), on the other hand, activates the data out only during transmission. In the momentary type, the data does not remain in the memory after the transmission is completed. Please refer to the diagram below:

d. OPERATION FLOWCHART:
**DECODER WITHOUT DATA OUTPUT PIN:**

1. When Power is turned on, PT2272 activates the Stand-By Mode.
2. It then searches for signals. If there is no signal received, it remains in the Stand-By Mode; otherwise, the address bits received are compared with the address configuration of the pins.
3. The VT goes high signifying the validation of transmission only when there are two (2) continuous frames that contain matched address bits; otherwise, VT will not be activated and the Stand-By Mode remains active.
4. Then, the Address Bits are again checked. Two continuous mismatches of the address bits would disable the VT and make the Stand-By Mode active; otherwise, the address bits are continuously checked.

![Decoder Diagram](image)

**DECODER WITH DATA OUTPUT PINS:**
1. When Power is turned ON, PT2272 activates the Stand-By Mode.
2. It then searches for signals. If there is no signal received, it remains in the Stand-By Mode; otherwise, the address bits are compared with the address configuration of the pins.
3. Whenever the Address Bits in a Frame match with that of the Address Configuration of the Pin, the data bits are stored into the memory. Also, when this IC finds two (2) continuous and identical data having the same address bits, the data output(s) is activated and the VT is enabled. The VT is disabled when there are 2 continuous mismatched addresses. For the momentary type, the data output is reset; while for the latch type, the data output is sustained.

3.2 DRIVER IC (L293D)
ICL293D is a motor drive IC. The four outputs from microcontroller are fed to the driver IC. It is L293D IC which drives the two motors of robot.

3.2.1 Features of L293D:

- 600mA. OUTPUT CURRENT CAPABILITY PER CHANNEL
- 1.2A PEAK OUTPUT CURRENT (NON REPETITIVE)PER CHANNEL
- ENABLE FACILITY
- OVERTEMPERATURE PROTECTION
- LOGICAL ”0” INPUT VOLTAGE UP TO 1.5v (HIGH NOISE IMMUNITY)
- INTERNALCLAMPS DIODE

DESCRIPTION

The L293D is a monolithic integrated high voltage, high current four channel driver designed to accept standard DTL or TTL logic levels and drive inductive loads (such as relays solenoids, DC and stepping motors) and switching power transistors. To simplify use as two bridges is pair of channels is equipped with an enable input. A separate supply input is provided form the logic, allowing operation at a low voltage and internal clamp diodes are included. The L293D is quadruple high-current half-H drivers. The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. Both devices are designed to drive inductive loads such as relays, solenoids, dc and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications.

3.2.2 BLOCK DIAGRAM
In the above block diagram drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN. When an enable input is high, the associated drivers are enabled and their outputs are active and in phase with their inputs. When the enable input is low, those drivers are disabled and their outputs are off and in the high-impedance state. With the proper data inputs, each pair of drivers forms a full-H (or bridge) reversible drive suitable for solenoid or motor applications. A VCC1 terminal, separate from VCC2, is provided for the logic inputs to minimize device power dissipation.
All inputs are TTL compatible. Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo-Darlington source as shown in the above schematic figure.

3.2.3 SPECIFICATIONS:

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

Supply voltage, VCC1 ................................................................. 36 V
Output supply voltage, VCC2 ......................................................... 36 V
Input voltage, VI ................................................................. 7 V
Output voltage range, VO ......................................................... –3 V to VCC2 + 3 V
Peak output current, IO (non repetitive, t ≤ 5 ms): L293 .............. ±2 A
Peak output current, IO (non repetitive, t ≤ 100 μs): L293D .............. ±1.2 A
Continuous output current, IO: L293 ...........................................±1 A
Continuous output current, IO: L293D ...........................................±600 mA
Continuous total dissipation at (or below) 25°C free-air temperature .......... 2075 mW
Continuous total dissipation at 80°C case temperature. .................5000 mW
Maximum junction temperature, TJ .............................................150°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds .......... 260°C
Storage temperature range, Tstg .................................................–65°C to 150°C
3.3 DETECTION CIRCUITS

The metal detector circuits are of various types. Here we use two metal detector circuits based on very low frequency (VLF) of operation. They are as follows.

METAL DETECTOR CIRCUIT:

The metal detector circuit is interfaced with the robo and it is left on the required search area in order to detect the metallic components used in the landmines.

The block diagram of the metal detector circuit is as shown in figure.

Figure 3.15: Block Diagram of Metal Detector Circuit

The main blocks of the metal detector circuit are
(1) Colpitts oscillator circuit.
(2) Sine to square wave generator circuit.
(3) Converting P-P voltage into 0-5v voltage.
(4) Programming the microcontroller timers to count the pulses.
(5) Alarm section

(1) OSCILLATOR CIRCUIT:

The oscillators which use the elements Land C to produce the oscillations are called LC oscillators. The circuit using L and C is called tank circuit or oscillatory circuit, which is an important part of oscillators. This circuit is also referred as resonating circuit, or tuned circuit.

The oscillator which is used in the metal detection circuit is a colpitts oscillator. An LC oscillator which uses two capacitive reactance and one inductive reactance in the feedback network i.e. tank circuit, is called colpitts oscillator. The amplifier stage uses an active device as a transistor in common emitter configuration. The circuit diagram of colpitts oscillator is as shown figure:
The resistances used across the base terminal of the transistor are biasing resistors. Inductor and two capacitors form the tank circuit. The colpitts oscillator generates the sinusoidal waves at a frequency of

\[
f_0 = \frac{1}{2\pi \sqrt{L \cdot \left(\frac{C_1 \cdot C_2}{C_1 + C_2}\right)}}
\]

The output of colpitts oscillator is given to the sine to square wave generator. The COLPITTS OSCILLATOR has fairly good frequency stability, is easy to tune, and can be used for a wide range of frequencies.

(2) **SINE TO SQUARE WAVE GENERATOR:**

The sinusoidal wave pulses are fed as input to the comparator circuit (sine to square wave generator) to convert into the square wave. The comparator is zero- crossing point detector of the input wave. The comparator will compare the two inputs. Whenever, the amplitudes of both the input signals are equal, transition takes place from high to low or low to high at the output of the comparator.

A zero-crossing detector delivers an output pulse that synchronizes other circuitry to the transitions through zero volts of a sinusoidal source for both polarity excursions. The circuit diagram is as shown in the figure.
In the comparator circuit, Op-amp IC741 is used to convert sinusoidal wave into the square wave. The pin diagram of the IC741 is as shown in the figure:
IC op-amps usually consist of four cascaded blocks. The first two stages are cascaded differential amplifiers used to provide high gain and high input resistance. The third stage acts as a buffer as well as a level shifter. The buffer is usually an emitter follower whose input impedance is very high so that it prevents loading of the high gain stage. The level shifter adjusts the D.C. voltages so that output voltage is zero for zero inputs. The adjustment of D.C. level is required as the gain stages are direct coupled usually. The output stage is designed to provide low output impedance as demanded by the ideal op-amp characteristics. The output voltage should swing symmetrically with respect to ground. To allow such symmetrical swing, the amplifier is provided with both positive and negative supply voltages.
3) **CONVERTING PEAK-PEAK VOLTAGE TO 0 TO 5V**

Transistor as a switch logic is used to convert the peak to peak voltage into 0 to 5 volts amplitude square wave. These converted pulses are fed to the microcontroller for counting the pulses.

The circuit to convert the peak to peak amplitude into 0 to 5 volts is as shown in the figure:

![Circuit diagram](image)

**Figure 3.18 Circuit converting peak to peak voltage to 0 to 5v**

Whenever a high voltage is applied at the input terminal i.e., greater than the biasing voltage of the transistor. The transistor comes into ON state. The voltage across the collector terminal is LOW voltage and across the emitter terminal is HIGH.

If the voltage at the input is low i.e., less than the biasing voltage of the transistor. The transistor is in OFF state. The voltage across the collector terminal is HIGH voltage and across the emitter is high.

As the input switches between the high and low, since square input is applied. Clock type pulses are generated across the output terminal.
(4) **PROGRAMMING IN MICRO CONTROLLER:**

The pulses from the transistor as a switch circuit are given to the MICROCONTROLLER Timer0. Timer0 is set as counter and Timer1 is set as timer using TMOD register. Timer0 and Timer1 is loaded with the initial values and set the run bits of two timers. Timer0 counts the external pulses. How much time should Timer0 count the pulses is kept in the Timer1. When overflow occurs in the Timer1, it executes Timer1 ISR.

The flow chart for the program in the MICROCONTROLLER is:
In Timer1 ISR, initial it halts the two timers and compares the value of Timer0 with the initial value. If timer0 value is less than initial value, no metal is detected. Otherwise, metal is detected.

If metal is detected then MICROCONTROLLER port1.4 pin is set high in order to activate the alarm. Otherwise, pin is set low indicating that no metal is detected to the alarm.
Figure 3.19 The circuit of metal detector
CHAPTER: 4
IMPLEMENTATION OF LAND MINE SWEEPER ROBO

4.1 DETECTION OF LAND MINE IN REAL TIME

In real time, the detection of landmines is done by a person using hand-held metal detector equipment as shown in figure. The person goes directly into the search area and performs the operation of searching the landmines.
Here, the person who is searching the landmines performing a risky job because unfortunately if the landmine explodes it leads to his death. So, in our project we are interfacing the metal detector circuit to a robot in order to search the landmines.

4.2 BLOCK DIAGRAM OF LAND MINE SWEEPER ROBO
In the above block diagram the RF transmitter and Receiver are used for controlling the Robo by wireless. The micro controller plays a vital role in our project. It controls the direction of Robo. It also activates the alarm section, whenever a metal is detected by Detection circuit.

4.3 DESIGN FLOW OF OUR PROJECT

The design flow of our project can be clearly explained by two sections. They are:

1) A Four wheeled Robo which can operate in two modes.
(i) Manual mode
(ii) Auto mode

2) Detection section.

**FOUR WHEELED ROBO:**

Four wheeled Robo is operated using the RF module. The RF receiver receives signals from the RF transmitter and gives it to the microcontroller as an input to it. Microcontroller checks whether to operate in manual mode or auto mode and performs the motion operation of the Robo, by driving the motors using the motor driver IC.

**RF Module:**

In order to control the movement of robo we are using the 4 channel RF module which described earlier. This module consists of the transmitter and the receiver section.

The transmitter section is in form of a key chain which consists of four switches. So, whenever a switch is pressed its corresponding code word generated by the Encoder IC SC2262 and is modulated by carrier.

Now this modulated signal is transmitted by Whip antenna and received by corresponding helical antenna. The received signal is again regenerated by the regenerator circuit and fed to the Decoder IC SC2272. It decodes the received codeword. The output is taken from the pins 13, 12, 11, 10 of decoder IC and is fed to micro controller IC AT89C51. It is as shown in the below figure.
Figure 4.1 RF module circuit

**Microcontroller:**

As we know that all the ports in this micro controller section bidirectional I/O ports, the outputs of Decoder IC are given to the port 1 of micro controller section as input. Based on the program written in it, it generates corresponding output from port 0. Here the port 0 is the output port.

**Driver IC (L293D):**

In order to drive the motors of robo we need high current. So we interface the micro controller and motors of robo by a driver IC (L293D). The four outputs from microcontroller are fed to the pins 2, 7, 10, 15 of driver IC. Earlier we saw its features which output a current up to 600mA. The input and output connections of driver IC are as shown in the below figure.
DETECTION SECTION:

Land mines definitely consist of metallic components. So, on detecting the metallic components in it by using a sensitive metal detector circuit, we can identify the landmines.

The Robo movement and metal detection circuit both switch on parallel. Whenever a land mine is detected by the detection circuit an alarm switch is activated. Simultaneously a signal is fed to microcontroller section as an interrupt and this interrupt stops the motion of Robo. In our project, we are using two metal detectors. One is placed in front of the Robo and other one is placed beside. So that by using these two metal detector circuits we can cover the more scanning area within a less time. The main advantage with this circuit is that we can make these detection circuits with less cost and having more efficiency.
CHAPTER: 5

MODES OF OPERATION

5.1 MANUAL MODE:
As we have mentioned earlier about the two modes of robot (Manual & Auto), we are going to deal with one of them now. It is Manual mode.

In manual mode initially the robot is left on the required search area and its motion is controlled by RF transmitter. Whenever the metal detector senses the metal the robot stops moving. Now the motion of the robot is again controlled by controller by switching a key in RF module in order to change its direction.

It can be clearly explained by the following flow chart. The input from RF receiver is taken and fed it to microcontroller section in which it first checks (like a switch condition in C language) which button is pressed. Then based on the correct condition it will follow the respective direction. It also checks the condition whether the metal is detected or not. If it detects then the motion of robot is stopped and alarm section is activated.

It will remain in same state until all the inputs are zero. When all the inputs are low and metal detect pin is high, then it will continue its motion according to the input code generated at the RF Transmitter and the alarm section is still in activate state, until metal detect pin is low. After the metal detect pin is low, it will go in to the normal operation.

**Flow chart for Manual Operation:**
5.2 AUTO MODE:

In Auto mode also the robot is placed in required search area. But based on the required condition written in microcontroller the robot is moved in three different directions automatically, for a predefined time. In our project, we programmed the robot motion according to the below path.

Robot enters into automatic mode, when MICROCONTROLLER receives high signals at three pins of RF receiver. Initially four variables are loaded with a predefined values.

1) One is for long forward direction (lon).
2) Second one for turning of robo (turn) for both left and right direction.
3) Third one for short forward path (shor).
4) Fourth one is for how much time the robo should be in auto mode (n).
**Long forward path direction:**

Here the variable is lon. Let us considered a value 500 is loaded in lon variable. Now the direction of robo is as shown in the below figure. While it is on its way, the lon variable is decremented. After reaching the lon value to zero it stops moving in forward direction.

![Moving Forward Diagram]

**Turning:**

Here the variable is turn. Let us considered a value 80 is loaded in lon variable. Now the direction of robo is as shown in the below figure. While it is on its way, the turn variable is decremented. After reaching the turn value to zero it stops turning. Here both left and right turns are also shown below.
**Short Forward path:**

Here the variable is Shor. Let us considered a value 200 is loaded in shor variable. Now the direction of robo is as shown in the below figure. While it is on its
way, the Shor variable is decremented. After reaching the Shor value to zero it stops moving in that direction.

![Diagram showing Shor variable decrement and auto mode of robo](image)

**Time duration for Auto mode of robo:**

Robo is in auto mode for a time interval, which is pre-defined in the program by using the variable ‘n’. The ‘n’ variable is decremented by ‘1’ for every two left turnings of robo. So, it internally checks the value whether it is ‘zero’ or not. If the value of ‘n’ is zero the robo stop its motion.
Flow chart of Auto Mode:

1. **Initialize Variables**
   - Lon
   - Turn
   - Short
   - n

2. **Decision**: If (n > 0)
   - **NO**
     - If (Lon > 0)
       - **NO**
         - If (Metal Detect = 1)
           - **YES** Forward
           - **NO** Lon Decrement
     - **YES**
       - If (Turn = 1)
         - **YES** Take Left Turn
         - **NO** Turn Decrement
   - **NO**
     - If (Short = 0)
       - **NO**
         - **YES**
         - **NO**

3. **Set alarm bit (a = 1)**

4. **Identification mark**

5. **STOP**
RESULTS

- Hence our Robot operated by a RF module can move in four directions i.e, forward, backward, left and right directions successfully.

- RF Module can be operated at a distance of 100m.

- Metals can be detected by using metal detector circuit. The metals can be detected at a range of 5 cms.
The current number of landmines deployed in the world is estimated at around 110 million. At the current rate of detection and clearing, we’re looking at over 500 years before all mines are cleared. It is imperative that more research is conducted in order to develop new methods to detect landmines, quickly, cheaply, and safely, and to also improve the sensitivity of current techniques.

Our Landmine Sweeper Robot, interfaced with two metal detectors, is widely used to scan a large area in less time and it is capable of detecting the landmines which are placed just below the ground surface.

By improving the sensitivity of Landmine Sweeper Robo, we can replace the military person who searches for the land mines. Hence we can save the life of a person in a real time.
➢ By providing artificial intelligence, it can detect and defuse (clear) the bombs automatically without any interference of others.

➢ By providing the water proof for the equipment in Robot, it can detect the underwater landmines also.

➢ An arm can be interfaced to the Robot in order to pick the objects and disposal of bombs.

➢ A laser guided gun is interfaced to the robot to use in defence.

➢ Proximity sensors are used in order to change the Robot direction automatically.
BIBLIOGRAPHY

- www.atmel.com
- www.wikipedia.com
- www.howstuffworks.com
- www.robot.com
APPENDIX

SOURCE CODE:

#include<reg51.h>

sbit pt=P1^1;     /*** detection of metal***/
sbit al=P1^2;     /*** signal to alarm ***/
sbit mark=P1^3;    /*** marking the metal***/
sbit a=P1^7;      /*** RF input1 ***/
sbit b=P1^6;      /*** RF input2 ***/
sbit c=P1^5;      /*** RF input3 ***/
sbit d=P1^4;      /*** RF input4 ***/

void automode();
void detect();
void main()
{
    /***Initialising timers ***/
    TMOD=0X15;
    TH0=0x98;
    TL0=0x58;
    TH1=0xf9;
    TL1=0xff;
    TR1=1;
    TR0=1;
    IE=0x08;
EA=1;

/*** initially zeroing the o/p port ***/
P0=0x00;
P1=0x00;
mark=0;
al=0;

/*** code for robo operation before detection **/
while(1)
{
    while(pt!=1)
    {
        if(a==1 & b==1 & c==1) /* checking auto mode or manual mode **/
            automode();
        else if(a==1)
            P0=0x50;
        else if(b==1)
            P0=0xa0;
        else if(c==1)
            P0=0x90;
        else if(d==1)
            P0=0x60;
        else
            P0=0x00;
    }
    while ((a | b | c | d)!==0)
{ 
    P0=0x00;
    al=1;         /* alarm starts **/
}
al=0;            /* after verification it stops sound**/

/** motoin of robo after detection **/
while(pt==1)
{
    if(a==1 & b==1 & c==1) /* checking auto mode or manual mode **/
        automode();
    else if(a==1)
        P0=0x50;
    else if(b==1)
        P0=0xa0;
    else if(c==1)
        P0=0x90;
    else if(d==1)
        P0=0x60;
    else
    {
        P0=0x00;
    }
}

/** isr for timer 1 over flow**/
void timer1(void) interrupt 3
TR1=0;
TR0=0;
if(TL0>5)
    pt=1;
else
    pt=0;

/*************reset the values**********/
TH0=0x98;
    TL0=0x58;
    TH1=0xf9;
    TL1=0xff;
    TR1=1;
    TR0=1;
return;
}

/*** program for auto mode***/

void automode()
{
    int n=10;
    int t1=500;
    int t2=50;
    int t=80;
    al=0;
    mark=0;
    while(n>0)
while(t1!=0)  /*** forward motion -- long length **/
{
  P0=0xaf;
  t1--;
  detect();
}

while(t2!=0)  /*** turn left **/
{
  P0=0x60;
  t2--;
  detect();
}

85
t2=50;
    while(t1!=0) //**forward motion long path **/
    {
        P0=0xaf;
        t1--;
        detect();
    }

    t1=500;
    while(t2!=0) //*** right turn**/
    {
        P0=0x90;
        t2--;
        detect();
    }

    t2=50;
    while(t!=0) //**forward motion --short path **/
    {
        P0=0xaf;
        t--;
        detect();
    }

    t=80;
    while(t2!=0) //** right turn**/
    {
        P0=0x90;
        t2--;
        detect();
    }

86
t2=50;

n--;

}
P0=0x00;
return;
}
/** function for detection checking**/

void detect()
{
    if(pt==1)
    {
        al=1;
        mark=1;
    }
    else
    {
        al=0;
        mark=0;
    }
    return;
}