CHAPTER 1
INTRODUCTION
1. INTRODUCTION

1.1 INTRODUCTION TO LEVEL CROSSING

What is a level crossing?

The place where track and highway/road intersects each other at the same level is known as “level crossing”.

There are mainly two types of level crossing on IR:

- Manned level crossing.
- Unmanned level crossing.

Classification of level crossing

Manned level crossing

- Spl.Class
- ‘A’ Class
- ‘B’ Class
- ‘C’ Class

Unmanned level crossing

- ‘C’ Class
- ‘D’ Class
1.2 COMPONENTS REQUIRED

Hardware

- Microcontroller(AT89C52)
- L293D
- Stepper motor
- Power supply
- IR Sensors

Software

Keil

- **Languages:** Embedded (Or) Assembly language

1.3 INTRODUCTION TO EMBEDDED SYSTEMS

**Definition:** Embedded System

“Any sort of device which includes a programmable computer, but itself is not intended to be a general-purpose computer”.

**What is an Embedded System?**

An embedded system is a microcontroller / microprocessor based system that is built to control or monitor the functions of equipment, machinery or plant.

Unlike a PC, embedded system is not designed to be programmed by an end user. Embedded system always runs a fixed application.

**Desktop vs. Embedded System**

In contrast to Desktops that performs a Variety of tasks an Embedded System performs a Single, Well-Defined Task.
The System has a Processor, Associated Peripherals, and Software for a Specific Purpose.

For example, in a Mobile Phone the Embedded Processor needs to Process Voice (to Send and Receive Speech Signals) as well as implement Communication Protocols.

The Hardware is Custom Built for the Specific Purpose.

**CROSS Compilers**

Another distinguishing feature of embedded software development is cross compilers.

Cross compilers are the ones, which runs on a machine based on one type of CPU and produces a machine instructions for a different kind of CPU.

**1.4 DIFFERENCE BETWEEN MICROCONTROLLER AND MICROPROCESSOR**

**Microcontroller**

- Dedicated CPU
- Memory is inbuilt.
- A Computer on a Chip.
- We have more instructions for bit addressable.
- We have I/O ports on the chip itself.

**Microprocessor**

- General purpose CPU.
- Memory is External.
- A Chip on a Computer.
- We have no instructions for bit addressable.
- We have no I/O ports on the chip itself.
CHAPTER 2
WORKING PRINCIPLE
2.1 BLOCK DIAGRAM
2.2 KIT DIAGRAM
2.3 HARDWARE IMPLEMENTATION

Micro Controller (89C52)

Totally 40-pin DIP package manufactured with CMOS Technology.

L293D (motor driver)

Racially L293D 16DIP /ULN 2003 IC is used to drive the stepper motor.

Stepper motor

This is used to open and close the gates automatically when it is rotated clock wise or anticlockwise direction.

Stepper motor requires 500m amps current, so use the uln2003 or L293D drivers to drive the stepper motor.

Proximity Sensor

Two IR sensor pairs (331,333) are used for transmitting and receiving signals.

SOFTWARE REQUIREMENTS

Keil

- Languages: Embedded (Or) Assembly language

2.4 ACCIDENT AVOIDENCE DETAILS

When the train arrives in a particular direction the transmitter IR senses and generates appropriate signal, then at the same time the receiver IR receives the signal and generates an interrupt.

When the interrupt is generated the stepper motor rotates in clockwise direction. When the interrupt ends the stepper motor rotates in anti clock wise direction.
CHAPTER 3
DESIGN PROCEDURE
MICROCONTROLLER
AT89C52
3.1 MICROCONTROLLER ARCHITECTURE (AT89C52)

3.1.1 GENERAL DESCRIPTION

The AT89C52 is a low-power, high-performance CMOS 8-bit microcomputer with 8K 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 80C51 and 80C52 instruction set and pin out.

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 AT89C52 is a powerful microcomputer, which provides a highly flexible and cost effective solution to many embedded control applications.

3.1.2 FEATURES OF MICROCONTROLLER

The AT89C52 provides the following standard features: 8K bytes of Flash, 256 bytes of RAM, 32 I/O lines, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89C52 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes.

The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power Down Mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next hardware reset.
3.1.3 PIN DIAGRAM

PDIP

(T2) P1.0  1  40  VCC
(T2 EX) P1.1  2  39  P0.0 (AD0)
P1.2  3  38  P0.1 (AD1)
P1.3  4  37  P0.2 (AD2)
P1.4  5  36  P0.3 (AD3)
P1.5  6  35  P0.4 (AD4)
P1.6  7  34  P0.5 (AD5)
P1.7  8  33  P0.6 (AD6)
RST  9  32  P0.7 (AD7)
(RXD) P3.0  10  31  EAVPP
(TXD) P3.1  11  30  ALE/PROG
(INT0) P3.2  12  29  PSEN
(INT1) P3.3  13  28  P2.7 (A15)
(T0) P3.4  14  27  P2.6 (A14)
(T1) P3.5  15  26  P2.5 (A13)
(WR) P3.6  16  25  P2.4 (A12)
(RD) P3.7  17  24  P2.3 (A11)
XTAL2  18  23  P2.2 (A10)
XTAL1  19  22  P2.1 (A9)
GND  20  21  P2.0 (A8)
3.1.4 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 can 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 pull-ups 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 pull-ups. In addition, P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively, as shown in the following table.
Port Pin Alternate Functions

P1.0 T2 (external count input to Timer/Counter 2), clock-out P1.1 T2EX (Timer/Counter 2 capture/reload trigger and direction control) Port 1 also receives the low-order address bytes during Flash programming and program verification.

Port 2

Port 2 is an 8-bit bi-directional I/O port with internal pull-ups. 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 pull-ups 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 pull-ups. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 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 pull-ups. 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 pull-ups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (IIL) because of the pull-ups. Port 3 also serves the functions of various special features of the AT89C51, as shown in the following table.
Port Pin Alternate Function

P3.0 RXD (serial input port)
P3.1 TXD (serial output port)
P3.2 INT0 (external interrupt 0)
P3.3 INT1 (external interrupt 1)
P3.4 T0 (timer 0 external input)
P3.5 T1 (timer 1 external input)
P3.6 WR (external data memory write strobe)
P3.7 RD (external data memory read strobe)

Port 3 also receives some control signals for Flash programming and programming 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 is an 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 micro controller is in external execution mode.
PSEN

Program Store Enable is the read strobe to external program memory. When the AT89C52 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 when 12-volt programming is selected.

XTAL1

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

XTAL2

Output from the inverting oscillator amplifier.

3.1.5 AT89C52 BLOCK DIAGRAM DESCRIPTION

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in Table 1. Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

User software should not write 1s to these unlisted locations, since they may be used in future products to invoke new features. In that case, the reset or inactive values of
the new bits will always be 0. Timer 2 Registers Control and status bits are contained in
registers T2CON (shown in Table 2) and T2MOD (shown in Table 4) for Timer 2.

The register pair (RCAP2H, RCAP2L) are the capture/Reload registers for Timer 2
in 16-bit capture mode or 16-bit auto-reload mode. Interrupt Registers the individual
interrupt enable bits are in the IE register. Two priorities can be set for each of the six-
interrupt sources in the IP register.
The AT89C52 implements 256 bytes of on-chip RAM. The upper 128 bytes occupy a parallel address space to the Special Function Registers. That means the upper 128 bytes have the same addresses as the SFR space but are physically separate from SFR space. When an instruction accesses an internal location above address 7FH, the address mode used in the instruction specifies whether the CPU accesses the upper 128 bytes of RAM or
the SFR space. Instructions that use direct addressing access SFR space. For example, the following direct addressing instruction accesses the SFR at location 0A0H (which is P2).

MOV 0A0H, #data

Instructions that use indirect addressing access the upper 128 bytes of RAM. For example, the following indirect addressing instruction, where R0 contains 0A0H, accesses the data byte at address 0A0H, rather than P2 (whose address is 0A0H).

MOV @R0, #data

Note that stack operations are examples of indirect addressing, so the upper 128 bytes of data RAM are available as stack space.

**Timer 0 and 1**

Timer 0 and Timer 1 in the AT89C52 operate the same way as Timer 0 and Timer 1 in the AT89C51.

**Timer 2**

Timer 2 is a 16-bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit C/T2 in the SFR T2CON (shown in Table 2). Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON, as shown in Table 3. Timer 2 consists of two 8-bit registers, TH2 and TL2. In the Timer function, the TL2 register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency. In the Counter function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T2.

In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since two machine cycles (24 oscillator periods) are required to recognize a 1-to-0 transition, the maximum count rate is
1/24 of the oscillator frequency. To ensure that a given level is sampled at least once before it changes, the level should be held for at least one full machine cycle.

<table>
<thead>
<tr>
<th>Table 3. Timer 2 Operating Modes</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCLK + TCLK</td>
</tr>
<tr>
<td>-------------</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>X</td>
</tr>
</tbody>
</table>

**Capture Mode**

In the capture mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 is a 16-bit timer or counter which upon overflow sets bit TF2 in T2CON. This bit can then be used to generate an interrupt. If EXEN2 = 1, Timer 2 performs the same operation, but a 1- to-0 transition at external input T2EX also causes the current value in TH2 and TL2 to be captured into RCAP2H and RCAP2L, respectively. In addition, the transition at T2EX Causes bit EXF2 in T2CON to be set. The EXF2 bit, like TF2, can generate an interrupt. The capture mode is illustrated in Figure 1.
Auto-reload (Up or Down Counter)

Timer 2 can be programmed to count up or down when configured in its 16-bit auto-reload mode. This feature is invoked by the DCEN (Down Counter Enable) bit located in the SFR T2MOD (see Table 4). Upon reset, the DCEN bit is set to 0 so that timer 2 will default to count up. When DCEN is set, Timer 2 can count up or down, depending on the value of the T2EX pin.

Figure 2 shows Timer 2 automatically counting up when DCEN = 0. In this mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to 0FFFFF and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16-bit value in RCAP2H and RCAP2L. The values in Timer in Capture ModeRCAP2H and RCAP2L are preset by software. If EXEN2 = 1, a 16-bit reload can be triggered either by an overflow or by a 1-to-0 transition at external input T2EX.

This transition also sets the EXF2 bit. Both the TF2 and EXF2 bits can generate an interrupt if enabled. Setting the DCEN bit enables Timer 2 to count up or down, as shown
in Figure 3. In this mode, the T2EX pin controls the direction of the count. Logic 1 at T2EX makes Timer 2 count up. The timer will overflow at 0FFFFH and set the TF2 bit. This overflow also causes the 16-bit value in RCAP2H and RCAP2L to be reloaded into the timer registers, TH2 and TL2, respectively.

Logic 0 at T2EX makes Timer 2 count down. The timer underflows when TH2 and TL2 equal the values stored in RCAP2H and RCAP2L. The underflow sets the TF2 bit and causes 0FFFFH to be reloaded into the timer registers. The EXF2 bit toggles whenever Timer 2 overflows or underflows and can be used as a 17th bit of resolution. In this operating mode, EXF2 does not flag an interrupt.
Automated Unmanned Level Railway Crossing System

Figure 3. Timer 2 Auto Reload Mode (DCEN = 1)

Figure 4. Timer 2 in Baud Rate Generator Mode
Baud Rate Generator

Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON (Table 2). Note that the baud rates for transmit and receive can be different if Timer 2 is used for the receiver or transmitter and Timer 1 is used for the other function. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 4. The baud rate generator mode is similar to the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2H and RCAP2L, which are preset by software. The baud rates in Modes 1 and 3 are determined by Timer 2’s overflow rate according to the following equation.

\[
\text{Modes 1 and 3 Baud Rates} = \frac{\text{Timer 2 Overflow Rate}}{16}
\]

The Timer can be configured for either timer or counter operation. In most applications, it is configured for timer operation (CP/T2 = 0). The timer operation is different for Timer 2 when it is used as a baud rate generator. Normally, as a timer, it increments every machine cycle (at 1/12 the oscillator frequency). As a baud rate generator, however, it increments every state time (at 1/2 the oscillator frequency). The baud rate formula is given below.

\[
\text{Modes 1 and 3 Baud Rate} = \frac{\text{Oscillator Frequency}}{32 \times [65536 - (\text{RCAP2H, RCAP2L})]}
\]

Where (RCAP2H, RCAP2L) is the content of RCAP2H and RCAP2L taken as a 16-bit unsigned integer. Timer 2 as a baud rate generator is shown in Figure 4. This figure is valid only if RCLK or TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2 and will not generate an interrupt. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt.
Note that when Timer 2 is running (TR2 = 1) as a timer in the baud rate generator mode, TH2 or TL2 should not be read from or written to. Under these conditions, the Timer is incremented every state time, and the results of a read or write may not be accurate. The RCAP2 registers may be read but should not be written to, because a write might overlap a reload and cause write and/or reload errors. The timer should be turned off (clear TR2) before accessing the Timer 2 or RCAP2 registers.
Figure 5. Timer 2 in Clock-out Mode

Programmable Clock Out

A 50% duty cycle clock can be programmed to come out on P1.0, as shown in Figure 5. This pin, besides being a regular I/O pin, has two alternate functions. It can be programmed to input the external clock for Timer/Counter 2 or to output a 50% duty cycle.
clock ranging from 61 Hz to 4 MHz at a 16 MHz operating frequency. To configure the Timer/Counter 2 as a clock generator, bit C/T2 (T2CON.1) must be cleared and bit T2OE (T2MOD.1) must be set. Bit TR2 (T2CON.2) starts and stops the timer. The clock-out frequency depends on the oscillator frequency and the reload value of Timer 2 captures registers (RCAP2H, RCAP2L), as shown in the following equation.

$$\text{Clock-Out Frequency} = \frac{\text{Oscillator Frequency}}{4 \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]}$$

In the clock-out mode, Timer 2 rollovers will not generate an interrupt. This behavior is similar to when Timer 2 is used as a baud-rate generator. It is possible to use Timer 2 as a baud-rate generator and a clock generator simultaneously. Note, however, that the baud-rate and clock-out frequencies cannot be determined independently from one another since they both use RCAP2H and RCAP2L.

**UART**

The UART in the AT89C52 operates the same way as the UART in the AT89C51.

**Interrupts**

The AT89C52 has a total of six interrupt vectors: two external interrupts (INT0 and INT1), three timer interrupts (Timers 0, 1, and 2), and the serial port interrupt. These interrupts are all shown in Figure 6. Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE. IE also contains a global disable bit, EA, which disables all interrupts at once. Note that Table shows that bit position IE.6 is unimplemented. In the AT89C51, bit position IE.5 is also unimplemented. User software should not write 1s to these bit positions, since they may be used in future AT89 products.
Timer 2 interrupt is generated by the logical OR of bits TF2 and EXF2 in register T2CON. Neither of these flags is cleared by hardware when the service routine is vectored to.

In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, and that bit will have to be cleared in software. The Timer 0 and Timer 1 flags, TF0 and TF1, are set at S5P2 of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle. However, the Timer 2 flag, TF2, is set at S2P2 and is polled in the same cycle in which the timer overflows.

**Figure 6. Interrupt Sources**

![Interrupt Sources Diagram]

**Oscillator Characteristics**

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 7. 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 8. 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.

### Programming the Flash

The AT89C52 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 AT89C52 inside the user’s system, while the high-voltage programming mode is compatible with conventional third party Flash or EPROM programmers. The AT89C52 is shipped with either the high-voltage or low-voltage programming mode enabled. The respective topside 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>AT89C52</td>
<td>AT89C52</td>
<td>AT89C52</td>
</tr>
<tr>
<td>xxxx</td>
<td>xxxx - 5</td>
<td>yyyy</td>
</tr>
<tr>
<td>yyyy</td>
<td>yyyy</td>
<td>yyyy</td>
</tr>
</tbody>
</table>

The AT89C52 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.
Programming Algorithm

Before programming the AT89C52, the address, data and control signals should be set up according to the Flash programming mode table and Figure 9 and Figure 10. To program the AT89C52, 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.

Data Polling

The AT89C52 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 data on PO.7. Once the write cycle has been completed, true data is 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 1s. The chip erase operation must be executed before the code memory can be reprogrammed.

**Reading the Signature Bytes**

The signature bytes are read by the same procedure as a normal verification of locations 030H, 031H, and 032H, except that P3.6 and P3.7 must be pulled to logic low. The values returned are as follows.

(030H) = 1EH indicates manufactured by Atmel
(031H) = 52H indicates 89C52
(032H) = FFH indicates 12V programming
(032H) = 05H indicates 5V programming
Automated Unmanned Level Railway Crossing System

Flash Programming and Verification Waveforms - High-voltage Mode (V_{PP}=12V)

Flash Programming and Verification Waveforms - Low-voltage Mode (V_{PP}=5V)
Automated Unmanned Level Railway Crossing System

**External Program Memory Read Cycle**

- **ALE**
- **PSEN**
- **PORT 0**
  - A0 - A7
  - INSTR IN
- **PORT 2**
  - A8 - A15

**External Data Memory Read Cycle**

- **ALE**
- **PSEN**
- **RD**
- **PORT 0**
  - A0 - A7 FROM P0 OR DP0
  - DATA IN
- **PORT 2**
  - F2.0 - F2.7 OR A8 - A15 FROM DPH
  - A8 - A15 FROM PCH
**External Data Memory Write Cycle**

- ALE
- PSEN
- WR

PORT 0: A0 - A7 FROM R1 OR DPL
DATA OUT
A0 - A7 FROM PCL
INSTR
IN

PORT 2: P2.0 - P2.7 OR A8 - A15 FROM DPH
A8 - A15 FROM PCH

**External Clock Drive Waveforms**

- $V_{DD} - 0.5V$
- $0.2V_{DD} - 0.1V$
- $0.7V_{DD}$

**External Clock Drive**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Min</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1/t_{CLOC}$</td>
<td>Oscillator Frequency</td>
<td>0</td>
<td>24</td>
<td>MHz</td>
</tr>
<tr>
<td>$t_{CLOC}$</td>
<td>Clock Period</td>
<td>41.6</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>$t_{CHCX}$</td>
<td>High Time</td>
<td>15</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>$t_{CLCX}$</td>
<td>Low Time</td>
<td>15</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>$t_{CRCH}$</td>
<td>Rise Time</td>
<td>20</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>$t_{CFCL}$</td>
<td>Fall Time</td>
<td>20</td>
<td></td>
<td>ns</td>
</tr>
</tbody>
</table>
## Serial Port Timing: Shift Register Mode Test Conditions

The values in this table are valid for $V_{CC} = 5.0V \pm 20\%$ and Load Capacitance = 80 pF.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>12 MHz Osc</th>
<th>Variable Oscillator</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{CLK}$</td>
<td>Serial Port Clock Cycle Time</td>
<td>Min 1.0</td>
<td>Max $12t_{CLC}$</td>
</tr>
<tr>
<td>$t_{DS}$</td>
<td>Output Data Setup to Clock Rising Edge</td>
<td>Min 700</td>
<td>Max $10t_{CLC}$-133</td>
</tr>
<tr>
<td>$t_{DH}$</td>
<td>Output Data Hold After Clock Rising Edge</td>
<td>Min 50</td>
<td>Max $2t_{CLC}$-117</td>
</tr>
<tr>
<td>$t_{HOLD}$</td>
<td>Input Data Hold After Clock Rising Edge</td>
<td>Min 0</td>
<td>Max 0</td>
</tr>
<tr>
<td>$t_{RVD}$</td>
<td>Clock Rising Edge to Input Data Valid</td>
<td>Min 700</td>
<td>Max $10t_{CLC}$-133</td>
</tr>
</tbody>
</table>

### Shift Register Mode Timing Waveforms

![Shift Register Mode Timing Waveforms](image)

### AC Testing Input/Output Waveforms

![AC Testing Input/Output Waveforms](image)

**Note:** 1. AC Inputs during testing are driven at $V_{CC} - 0.5V$ for a logic 1 and 0.45V for a logic 0. Timing measurements are made at $V_{CC}$ min. for a logic 1 and $V_{CC}$ max. for a logic 0.

### Float Waveforms

![Float Waveforms](image)

**Note:** 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when a 100 mV change from the loaded $V_{OH}/V_{OL}$ level occurs.
STEPPER MOTOR
3.2 STEPPER MOTOR

Basic Stepping Motor Control Circuits

- Introduction
- Variable Reluctance Motors
- Unipolar Permanent Magnet and Hybrid Motors
- Practical Unipolar and Variable Reluctance Drivers
- Bipolar Motors and H-Bridges
- Practical Bipolar Drive Circuits

3.2.1 Introduction

This section of the stepper tutorial deals with the basic final stage drive circuitry for stepping motors. This circuitry is centered on a single issue, switching the current in each motor winding on and off, and controlling its direction. The circuitry discussed in this section is connected directly to the motor windings and the motor power supply, and this circuitry is controlled by a digital system that determines when the switches are turned on or off.

This section covers all types of motors, from the elementary circuitry needed to control a variable reluctance motor, to the H-bridge circuitry needed to control a bipolar permanent magnet motor. Each class of drive circuit is illustrated with practical examples, but these examples are not intended as an exhaustive catalog of the commercially available control circuits, nor is the information given here intended to substitute for the information found on the manufacturer's component data sheets for the parts mentioned.

This section only covers the most elementary control circuitry for each class of motor. All of these circuits assume that the motor power supply provides a drive voltage no
greater than the motor’s rated voltage, and this significantly limits motor performance. The next section, on current limited drive circuitry, covers practical high-performance drive circuits.

3.2.2 Variable Reluctance Motors

Typical controllers for variable reluctance stepping motors are variations on the outline shown in Figure 3.1:

![Figure 3.1](image)

In Figure 3.1, boxes are used to represent switches; a control unit, not shown, is responsible for providing the control signals to open and close the switches at the appropriate times in order to spin the motors. In many cases, the control unit will be a computer or programmable interface controller, with software directly generating the outputs needed to control the switches, but in other cases, additional control circuitry is introduced, sometimes gratuitously.

Motor windings, solenoids and similar devices are all inductive loads. As such, the current through the motor winding cannot be turned on or off instantaneously without involving infinite voltages! When the switch controlling a motor winding is closed, allowing current to flow, the result of this is a slow rise in current. When the switch controlling a motor winding is opened, the result of this is a voltage spike that can seriously damage the switch unless care is taken to deal with it appropriately.
There are two basic ways of dealing with this voltage spike. One is to bridge the motor winding with a diode, and the other is to bridge the motor winding with a capacitor. Figure 3.2 illustrates both approaches:

![Figure 3.2](image)

The diode shown in Figure 3.2 must be able to conduct the full current through the motor winding, but it will only conduct briefly each time the switch is turned off, as the current through the winding decays. If relatively slow diodes such as the common 1N400X family are used together with a fast switch, it may be necessary to add a small capacitor in parallel with the diode.

The capacitor shown in Figure 3.2 poses more complex design problems! When the switch is closed, the capacitor will discharge through the switch to ground, and the switch must be able to handle this brief spike of discharge current. A resistor in series with the capacitor or in series with the power supply will limit this current. When the switch is opened, the stored energy in the motor winding will charge the capacitor up to a voltage significantly above the supply voltage, and the switch must be able to tolerate this voltage. To solve for the size of the capacitor, we equate the two formulas for the stored energy in a resonant circuit:

\[ P = C V^2 / 2 \]
\[ P = L \dot{I}^2 / 2 \]

Where:
- \( P \) -- stored energy, in watt seconds or coulomb volts
- \( C \) -- capacity, in farads
- \( V \) -- voltage across capacitor
\( L \) -- inductance of motor winding, in henrys  
\( I \) -- current through motor winding  
Solving for the minimum size of capacitor required to prevent overvoltage on the switch is fairly easy:  
\[ C > \frac{L I^2}{(V_b - V_s)^2} \]  
Where:  
\( V_b \) -- the breakdown voltage of the switch  
\( V_s \) -- the supply voltage

Variable reluctance motors have variable inductance that depends on the shaft angle. Therefore, worst-case design must be used to select the capacitor. Furthermore, motor inductances are frequently poorly documented, if at all.

The capacitor and motor winding, in combination, form a resonant circuit. If the control system drives the motor at frequencies near the resonant frequency of this circuit, the motor current through the motor windings, and therefore, the torque exerted by the motor, will be quite different from the steady state torque at the nominal operating voltage! The resonant frequency is:

\[ f = \frac{1}{2\pi \sqrt{L C}} \]

Again, the electrical resonant frequency for a variable reluctance motor will depend on shaft angle! When a variable reluctance motors is operated with the exciting pulses near resonance, the oscillating current in the motor winding will lead to a magnetic field that goes to zero at twice the resonant frequency, and this can severely reduce the available torque.
3.2.3 Unipolar Permanent Magnet and Hybrid Motors

Typical controllers for unipolar stepping motors are variations on the outline shown in Figure 3.3:

![Figure 3.3](image)

In Figure 3.3, as in Figure 3.1, boxes are used to represent switches; a control unit, not shown, is responsible for providing the control signals to open and close the switches at the appropriate times in order to spin the motors. The control unit is commonly a computer or programmable interface controller, with software directly generating the outputs needed to control the switches.

As with drive circuitry for variable reluctance motors, we must deal with the inductive kick produced when each of these switches is turned off. Again, we may shunt the inductive kick using diodes, but now, 4 diodes are required, as shown in Figure 3.4:

![Figure 3.4](image)

The extra diodes are required because the motor winding is not two independent inductors, it is a single center-tapped inductor with the center tap at a fixed voltage. This acts as an autotransformer! When one end of the motor winding is pulled down, the other end will fly up, and visa versa. When a switch opens, the inductive kickback will drive that
end of the motor winding to the positive supply, where it is clamped by the diode. The opposite end will fly downward, and if it was not floating at the supply voltage at the time, it will fall below ground, reversing the voltage across the switch at that end. Some switches are immune to such reversals, but others can be seriously damaged.

A capacitor may also be used to limit the kickback voltage, as shown in Figure 3.5:

Figure 3.5

![Diagram showing capacitor and control signals](image)

The rules for sizing the capacitor shown in Figure 3.5 are the same as the rules for sizing the capacitor shown in Figure 3.2, but the effect of resonance is quite different! With a permanent magnet motor, if the capacitor is driven at or near the resonant frequency, the torque will increase to as much as twice the low-speed torque! The resulting torque versus speed curve may be quite complex, as illustrated in Figure 3.6:

Figure 3.6

![Graph showing electrical and mechanical resonant speeds](image)

Figure 3.6 shows a peak in the available torque at the electrical resonant frequency, and a valley at the mechanical resonant frequency. If the electrical resonant frequency is
placed appropriately above what would have been the cutoff speed for the motor using a diode-based driver, the effect can be a considerable increase in the effective cutoff speed.

The mechanical resonant frequency depends on the torque, so if the mechanical resonant frequency is anywhere near the electrical resonance, it will be shifted by the electrical resonance! Furthermore, the width of the mechanical resonance depends on the local slope of the torque versus speed curve; if the torque drops with speed, the mechanical resonance will be sharper, while if the torque climbs with speed, it will be broader or even split into multiple resonant frequencies.

### 3.2.4 Practical Unipolar and Variable Reluctance Drivers

In the above circuits, the details of the necessary switches have been deliberately ignored. Any switching technology, from toggle switches to power MOSFETs will work! Figure 3.7 contains some suggestions for implementing each switch, with a motor winding and protection diode included for orientation purposes:

![Figure 3.7](image)

Each of the switches shown in Figure 3.7 is compatible with a TTL input. The 5 volt supply used for the logic, including the 7407 open-collector driver used in the figure, should be well regulated. The motor power, typically between 5 and 24 volts, needs only minimal regulation. It is worth noting that these power switching circuits are appropriate for driving solenoids, DC motors and other inductive loads as well as for driving stepping motors.
The SK3180 transistor shown in Figure 3.7 is a power darlington with a current gain over 1000; thus, the 10 milliamps flowing through the 470 ohm bias resistor is more than enough to allow the transistor to switch a few amps current through the motor winding. The 7407 buffer used to drive the darlington may be replaced with any high-voltage open collector chip that can sink at least 10 milliamps. In the event that the transistor fails, the high-voltage open collector driver serves to protect the rest of the logic circuitry from the motor power supply.

The IRC IRL540 shown in Figure 3.7 is a power field effect transistor. This can handle currents of up to about 20 amps, and it breaks down nondestructively at 100 volts; as a result, this chip can absorb inductive spikes without protection diodes if it is attached to a large enough heat sink. This transistor has a very fast switching time, so the protection diodes must be comparably fast or bypassed by small capacitors. This is particularly essential with the diodes used to protect the transistor against reverse bias! In the event that the transistor fails, the zener diode and 100 ohm resistor protect the TTL circuitry. The 100 ohm resistor also acts to somewhat slow the switching times on the transistor.

For applications where each motor winding draws under 500 milliamps, the ULN200x family of darlington arrays from Allegro Microsystems, also available as the DS200x from National Semiconductor and as the Motorola MC1413 darlington array will drive multiple motor windings or other inductive loads directly from logic inputs. Figure 3.8 shows the pin out of the widely available ULN2003 chip, an array of 7 darlington transistors with TTL compatible inputs:

![ULN2003 Pin Out Diagram](image)
The base resistor on each Darlington transistor is matched to standard bipolar TTL outputs. Each NPN Darlington is wired with its emitter connected to pin 8, intended as a ground pin. Each transistor in this package is protected by two diodes, one shorting the emitter to the collector, protecting against reverse voltages across the transistor, and one connecting the collector to pin 9; if pin 9 is wired to the positive motor supply, this diode will protect the transistor against inductive spikes.

The ULN2803 chip is essentially the same as the ULN2003 chip described above, except that it is in an 18-pin package, and contains 8 darlington, allowing one chip to be used to drive a pair of common unipolar permanent-magnet or variable-reluctance motors.

For motors drawing under 600 milliamps per winding, the UDN2547B quad power driver made by Allegro Microsystems will handle all 4 windings of common unipolar stepping motors. For motors drawing under 300 milliamps per winding, Texas Instruments SN7541, 7542 and 7543 dual power drivers are a good choice; both of these alternatives include some logic with the power drivers.

**3.2.5 Bipolar Motors and H-Bridges**

Things are more complex for bipolar permanent magnet stepping motors because these have no center taps on their windings. Therefore, to reverse the direction of the field produced by a motor winding, we need to reverse the current through the winding. We could use a double-pole double throw switch to do this electromechanically; the electronic equivalent of such a switch is called an H-bridge and is outlined in Figure 3.9:
As with the unipolar drive circuits discussed previously, the switches used in the H-bridge must be protected from the voltage spikes caused by turning the power off in a motor winding. This is usually done with diodes, as shown in Figure 3.9.

It is worth noting that H-bridges are applicable not only to the control of bipolar stepping motors, but also to the control of DC motors, push-pull solenoids (those with permanent magnet plungers) and many other applications.

With 4 switches, the basic H-bridge offers 16 possible operating modes, 7 of which short out the power supply! The following operating modes are of interest:

*Forward mode*, switches A and D closed.

*Reverse mode*, switches B and C closed.

These are the usual operating modes, allowing current to flow from the supply, through the motor winding and onward to ground. Figure 3.10 illustrates forward mode:

*Fast decay mode or coasting mode*, all switches open.
Any current flowing through the motor winding will be working against the full supply voltage, plus two diode drops, so current will decay quickly. This mode provides little or no dynamic braking effect on the motor rotor, so the rotor will coast freely if all motor windings are powered in this mode. Figure 3.11 illustrates the current flow immediately after switching from forward running mode to fast decay mode.

Figure 3.11
*Slow decay modes or dynamic braking modes.*

In these modes, current may recirculate through the motor winding with minimum resistance. As a result, if current is flowing in a motor winding when one of these modes is entered, the current will decay slowly, and if the motor rotor is turning, it will induce a current that will act as a brake on the rotor. Figure 3.12 illustrates one of the many useful slow-decay modes, with switch D closed; if the motor winding has recently been in forward running mode, the state of switch B may be either open or closed:

Figure 3.12

Most H-bridges are designed so that the logic necessary to prevent a short circuit is included at a very low level in the design. Figure 3.13 illustrates what is probably the best arrangement:
Here, the following operating modes are available:

<table>
<thead>
<tr>
<th>XY</th>
<th>ABCD</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0000</td>
<td>fast decay</td>
</tr>
<tr>
<td>01</td>
<td>1001</td>
<td>forward</td>
</tr>
<tr>
<td>10</td>
<td>0110</td>
<td>reverse</td>
</tr>
<tr>
<td>11</td>
<td>0101</td>
<td>slow decay</td>
</tr>
</tbody>
</table>

The advantage of this arrangement is that all of the useful operating modes are preserved, and they are encoded with a minimum number of bits; the latter is important when using a microcontroller or computer system to drive the H-bridge because many such systems have only limited numbers of bits available for parallel output. Sadly, few of the integrated H-bridge chips on the market have such a simple control scheme.

3.2.6 Practical Bipolar Drive Circuits

There are a number of integrated H-bridge drivers on the market, but it is still useful to look at discrete component implementations for an understanding of how an H-bridge works. Antonio Raposo suggested the H-bridge circuit shown in Figure 3.14;
The X and Y inputs to this circuit can be driven by open collector TTL outputs as in the Darlington-based unipolar drive circuit in Figure 3.7. The motor winding will be energized if exactly one of the X and Y inputs is high and exactly one of them is low. If both are low, both pull-down transistors will be off. If both are high, both pull-up transistors will be off. As a result, this simple circuit puts the motor in dynamic braking mode in both the 11 and 00 states, and does not offer a coasting mode.

The circuit in Figure 3.14 consists of two identical halves, each of which may be properly described as a push-pull driver. The term half H-bridge is sometimes applied to these circuits! It is also worth noting that a half H-bridge has a circuit quite similar to the output drive circuit used in TTL logic. In fact, TTL tri-state line drivers such as the 74LS125A and the 74LS244 can be used as half H-bridges for small loads, as illustrated in Figure 3.15:
This circuit is effective for driving motors with up to about 50 ohms per winding at voltages up to about 4.5 volts using a 5 volt supply. Each tri-state buffer in the LS244 can sink about twice the current it can source, and the internal resistance of the buffers is sufficient, when sourcing current, to evenly divide the current between the drivers that are run in parallel. This motor drive allows for all of the useful states achieved by the driver in Figure 3.13, but these states are not encoded as efficiently:

<table>
<thead>
<tr>
<th>XYE</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>110</td>
<td>slow decay</td>
</tr>
<tr>
<td>100</td>
<td>reverse</td>
</tr>
<tr>
<td>010</td>
<td>forward</td>
</tr>
<tr>
<td>000</td>
<td>slower decay</td>
</tr>
<tr>
<td>--1</td>
<td>fast decay</td>
</tr>
</tbody>
</table>

The second dynamic braking mode, XYE=110, provides a slightly weaker braking effect than the first because of the fact that the LS244 drivers can sink more current than they can source.

The Microchip (formerly Telecom Semiconductor) TC4467 Quad CMOS driver is another example of a general purpose driver that can be used as 4 independent half H-bridges. Unlike earlier drivers, the data sheet for this driver even suggests using it for motor control applications, with supply voltages up to 18 volts and up to 250 milliamps per motor winding.

One of the problems with commercially available stepping motor control chips is that many of them have relatively short market lifetimes. For example, the Seagate IPxMxx series of dual H-bridge chips (IP1M10 through IP3M12) were very well thought out, but unfortunately, it appears that Seagate only made these when they used stepping motors for head positioning in Seagate disk drives.
The Toshiba TA7279 dual H-bridge driver would be another excellent choice for motors under 1 amp, but again, it appears to have been made for internal use only.

The SGS-Thompson (and others) L293 dual H-bridge is a close competitor for the above chips, but unlike them, it does not include protection diodes. The L293D chip, introduced later, is pin compatible and includes these diodes. If the earlier L293 is used, each motor winding must be set across a bridge rectifier (1N4001 equivalent).

The use of external diodes allows a series resistor to be put in the current recirculation path to speed the decay of the current in a motor winding when it is turned off; this may be desirable in some applications. The L293 family offers excellent choices for driving small bipolar steppers drawing up to one amp per motor winding at up to 36 volts. Figure 3.16 shows the pinout common to the L293B and L293D chips:

![Figure 3.16](image)

This chip may be viewed as 4 independent half H-bridges, enabled in pairs, or as two full H-bridges. This is a power DIP package, with pins 4, 5, 12 and 13 designed to conduct heat to the PC board or to an external heat sink.

The SGS-Thompson (and others) L298 dual H-bridge is quite similar to the above, but is able to handle up to 2-amps per channel and is packaged as a power component; as with the LS244, it is safe to wire the two H-bridges in the L298 package into one 4-amp H-bridge (the data sheet for this chip provides specific advice on how to do this).

S.R.T.I.S.T. 53
One warning is appropriate concerning the L298; this chip very fast switches, fast enough that commonplace protection diodes (1N400X equivalent) don't work. Instead, use a diode such as the BYV27. The National Semiconductor LMD18200 H-bridge is another good example; this handles up to 3 amps and has integral protection diodes.

While integrated H-bridges are not available for very high currents or very high voltages, there are well designed components on the market to simplify the construction of H-bridges from discrete switches. For example, International Rectifier sells a line of half H-bridge drivers; two of these chips plus 4 MOSFET switching transistors suffice to build an H-bridge.

The IR2101, IR2102 and IR2103 are basic half H-bridge drivers. Each of these chips has 2 logic inputs to directly control the two switching transistors on one leg of an H-bridge. The IR2104 and IR2111 have similar output-side logic for controlling the switches of an H-bridge, but they also include input-side logic that, in some applications, may reduce the need for external logic. In particular, the 2104 includes an enable input, so that 4 2104 chips plus 8 switching transistors can replace an L293 with no need for additional logic.

The data sheet for the Microchip (formerly Telecom Semiconductor) TC4467 family of quad CMOS drivers includes information on how to use drivers in this family to drive the power MOSFETs of H-bridges running at up to 15 volts.

A number of manufacturers make complex H-bridge chips that include current limiting circuitry; these are the subject of the next section. It is also worth noting that there are a number of 3-phase bridge drivers on the market, appropriate for driving Y or delta configured 3-phase permanent magnet steppers.
Few such motors are available, and these chips were not developed with steppers in mind. Nonetheless, the Toshiba TA7288P, the GL7438, the TA8400 and TA8405 are clean designs, and 2 such chips, with one of the 6 half-bridges ignored, will cleanly control a 5-winding 10 step per revolution motor.
L293D
PUSH-PULL FOUR CHANNEL DRIVER WITH DIODES

3.3 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 1.5 V
### 3.3.1 DESCRIPTION

The Device 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 each pair of channels is equipped with an enable input. A separate supply input is provided for the logic, allowing operation at a lower voltage and internal clamp diodes are included.

This device is suitable for use in switching applications at frequencies up to 5 kHz. The L293D is assembled in a 16 lead plastic package which has 4 center pins connected together and used for heat sinking. The L293DD is assembled in a 20 lead surface mount which has 8 center pins connected together and used for heat sinking.
3.3.2 BLOCK DIAGRAM

PIN CONNECTIONS

ENABLE 1 1 20 Vss
INPUT 1 2 19 INPUT 4
OUTPUT 1 3 18 OUTPUT 4
GND 4 17 GND
GND 5 16 GND
GND 6 15 GND
GND 7 14 GND
OUTPUT 2 8 13 OUTPUT 3
INPUT 2 9 12 INPUT 3
Vs 10 11 ENABLE 2
3.3.3 ASSEMBLY INSTRUCTIONS

- Check the components supplied in the Kit against the Component list and identify all the components.
- It is generally best to solder the lowest height components first. Solder the components in the following order:
  - Jumpers, resistors, diodes, IC base, transistors and other components.
  - Take care of terminals polarity while soldering diodes, LED s and electrolytic capacitors.
  - Identify the terminals of transistors and solder them in correct direction.
  - Connect the LCD with the help of 16-pin male-female work-strip connector.
  - Use flux cored lead to avoid dry solderability.
  - Inspect the solder points against dry solder / excess solder
  - Now insert the preprogrammed microcontroller in the IC base firmly.
  - Adjust the 10K preset (near LCD) to correct contrast display level of LCD.
  - Ensure that the ac voltage to the kit is 12V. For this, use 230/12V step down transformer with 500mA or more current rating.
  - Use heat sink for voltage regulator, if required.
3.3.4 ADVANTAGES AND DISADVANTAGES OF L293D

ADVANTAGES

- Efficient way of speed control of DC motor.
- Produces more torque.
- Produces less noise.

DISADVANTAGES

- It is not applicable for AC motors

APPLICATIONS

- Industries.
- Traction.
- Home appliance.
REGULATED POWER SUPPLY
3.4 REGULATED POWER SUPPLY

3.4.1 DESCRIPTION

A variable regulated power supply, also called a variable bench power supply, is one where you can continuously adjust the output voltage to your requirements. Varying the output of the power supply is the recommended way to test a project after having double checked parts placement against circuit drawings and the parts placement guide.

This type of regulation is ideal for having a simple variable bench power supply. Actually this is quite important because one of the first projects a hobbyist should undertake is the construction of a variable regulated power supply. While a dedicated supply is quite handy e.g. 5V or 12V, it's much handier to have a variable supply on hand, especially for testing.

Most digital logic circuits and processors need a 5 volt power supply. To use these parts we need to build a regulated 5 volt source. Usually you start with an unregulated power supply ranging from 9 volts to 24 volts DC (A 12 volt power supply is included with the Beginner Kit and the Microcontroller Beginner Kit.). To make a 5 volt power supply, we use a LM7805 voltage regulator IC (Integrated Circuit). The IC is shown below.

![LM7805 IC](image)

The LM7805 is simple to use. You simply connect the positive lead of your unregulated DC power supply (anything from 9VDC to 24VDC) to the Input pin,
the negative lead to the Common pin and then when you turn on the power, you get a 5 volt supply from the Output pin.

3.4.2 **CIRCUIT FEATURES**

- **Brief description of operation:** Gives out well regulated +5V output, output current capability of 100 mA
- **Circuit protection:** Built-in overheating protection shuts down output when regulator IC gets too hot
- **Circuit complexity:** Very simple and easy to build
- **Circuit performance:** Very stable +5V output voltage, reliable operation
- **Availability of components:** Easy to get, uses only very common basic components
- **Design testing:** Based on datasheet example circuit, I have used this circuit successfully as part of many electronics projects
- **Applications:** Part of electronics devices, small laboratory power supply
- **Power supply voltage:** Un regulated DC 8-18V power supply
- **Power supply current:** Needed output current + 5 mA
- **Component costs:** Few dollars for the electronics components + the input transformer cost
3.4.3 BLOCK DIAGRAM

3.4.4 CIRCUIT DIAGRAM
TRANSFORMER

Usually DC voltages required for the operation of various electronic equipment is 6V, 9V or 12V. This voltage is quite small than the A.C. mains voltage. Therefore before rectification a step down transformer is employed to decrease the voltage to the required level.

RECTIFIER

It converts A.C. into pulsating D.C. the rectifier may be half wave or full wave. Mostly bridge type full wave rectifier is preferred because of its unchallenging merits.

FILTER

It removes the ripples from the output of rectifier and smoothens the D.C. Output received from the filter is constant till the mains voltage and load is kept constant. However, if either of the two is varied D.C. Voltage received at this point changes therefore a regulator is applied at the output stage.

REGULATOR

Eliminates ripple by setting DC output to a fixed voltage.

- Regulator
3.4.5 LIMITATIONS

The DC output voltage changes with a change in AC supply voltage. For example if an A.C. voltage decreases by 6% then DC voltage also decreases by 6%. The DC output voltage decreases considerably with the increase in load due to voltage drop in transformer, rectifier and filter circuits. These variations in DC output may affect the performance of electronic circuits.

For instance the frequency of oscillation will change in an oscillator and in transmitter the output may be destroyed. Thus it is concluded that ordinary power supply is not suitable for some of the electronic circuits for instance, with such circuits we have to employ a regulated power supply, which gives a fixed output.

3.4.6 NECESSITY OF REGULATED POWER SUPPLY

The DC level of an ordinary power supply changes due to the following reasons. Variations in AC mains voltage: the permissible variation in the mains voltage as per Indian electricity rules is +/- 6% of its rated value. But in India the variation in voltage is much more than its rated value.

That is why the DC voltage of an ordinary power supply changes to such an extent that the electronic device refuses to work satisfactorily. Voltages drop in internal resistance: the internal resistance of an ordinary power supply is very large. Therefore output voltage changes to an extent when load is connected across it. Some it reduces to very low extent due to which the electronic component refuses to work.

3.4.7 IC 7805 & IC 7812

7805 & 7812 is an integrated three-terminal positive fixed linear voltage regulator. It supports an input voltage of 10 volts to 35 volts and output voltage of 5 volts. It has a current rating of 1 amp although lower current models are available. Its output voltage is fixed at 5.0V. The 7805 also has a built-in current limiter as a safety feature. 7805 is
manufactured by many companies, including National Semiconductors and Fairchild Semiconductors.

The 7805 will automatically reduce output current if it gets too hot. The last two digits represent the voltage; for instance, the 7812 is a 12-volt regulator. The 78xx series of regulators is designed to work in complement with the 79xx series of negative voltage regulators in systems that provide both positive and negative regulated voltages, since the 78xx series can't regulate negative voltages in such a system.

The 7805 & 7812 is one of the most common and well-known of the 78xx series regulators, as it's small component count and medium-power regulated 5V make it useful for powering TTL devices.

<table>
<thead>
<tr>
<th>SPECIFICATIONS</th>
<th>IC 7805</th>
<th>IC 7812</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{out}$</td>
<td>5V</td>
<td>12V</td>
</tr>
<tr>
<td>$V_{cin} - V_{out}$ Difference</td>
<td>5V - 20V</td>
<td>5V – 20V</td>
</tr>
<tr>
<td>Operation Ambient Temp</td>
<td>0 - 125°C</td>
<td>0 - 125°C</td>
</tr>
<tr>
<td>Output $I_{max}$</td>
<td>1A</td>
<td>1A</td>
</tr>
</tbody>
</table>
3.5 IR SENSORS

Two IR sensor pairs (331,333) are used for transmitting and receiving signals.

IR CIRCUITS

This circuit has two stages: a transmitter unit and a receiver unit. The transmitter unit consists of an infrared LED and its associated circuitry.

IR TRANSMITTER

The transmitter circuit consists of the following components:

1. Resistors
2. IR LED

The IR LED emitting infrared light is put on in the transmitting unit. Infrared LED is driven through transistor BC 548.

Transmitter

```
  +---------+  
  |   VCC   |  
  +---------+  
     ↘  
     |  
     |  
     |  
     ↓  
100 ohms
```
IR RECEIVER

The receiver circuit consists of the following components:

1. Resistors.
2. IR LED.

The receiver unit consists of a sensor and its associated circuitry. In receiver section, the first part is a sensor, which detects IR pulses transmitted by IR-LED. Whenever a train crosses the sensor, the output of IR sensor momentarily transits through a low state.

As a result the monostable is triggered and a short pulse is applied to the port pin of the 8051 microcontroller. On receiving a pulse from the sensor circuit, the controller activates the circuitry required for closing and opening of the gates and for track switching. The IR receiver circuit is shown in the figure below.
CHAPTER 4
RESULTS OF THE PROJECT
4.1 AUTOMATIC RAILWAY GATE CONTROL

INTRODUCTION

Description:

Present project is designed using 8051 microcontroller to avoid railway accidents happening at unattended railway gates, if implemented in spirit. This project utilizes two powerful IR transmitters and two receivers; one pair of transmitter and receiver is fixed at up side (from where the train comes) at a level higher than a human being in exact alignment and similarly the other pair is fixed at down side of the train direction. Sensor activation time is so adjusted by calculating the time taken at a certain speed to cross at least one compartment of standard minimum size of the Indian railway.

We have considered 5 seconds for this project. Sensors are fixed at 1km on both sides of the gate. We call the sensor along the train direction as ‘foreside sensor’ and the other as ‘aft side sensor’.

When foreside receiver gets activated, the gate motor is turned on in one direction and the gate is closed and stays closed until the train crosses the gate and reaches aft side sensors.

When aft side receiver gets activated motor turns in opposite direction and gate opens and motor stops. Buzzer will immediately sound at the fore side receiver activation and gate will close after 5 seconds, so giving time to drivers to clear gate area in order to avoid trapping between the gates and stop sound after the train. Railways being the cheapest mode of transportation are preferred over all the other means. When we go through the daily newspapers we come across many railway accidents occurring at unmanned railway crossings.
This is mainly due to the carelessness in manual operations or lack of workers. We, in this project has come up with a solution for the same. Using simple electronic components we have tried to automate the control of railway gates. As a train approaches the railway crossing from either side, the sensors placed at a certain distance from the gate detects the approaching train and accordingly controls the operation of the gate. Also an indicator light has been provided to alert the motorists about the approaching train.

- Railway Crossing
Railway Track
4.2 Stepping Sequences for a Four-Phase Unipolar Permanent Magnet Stepper Motor

This kind of motor has four coils which, when energised in the correct sequence, cause the permanent magnet attached to the shaft to rotate.

There are two basic step sequences. After step 4, the sequence is repeated from step 1 again.

Reversing the order of the steps in a sequence will reverse the direction of rotation.

Here are some possible connection diagrams and some software

A. Single-Coil Excitation - Each successive coil is energized in turn.

<table>
<thead>
<tr>
<th>Step</th>
<th>Coil 4</th>
<th>Coil 3</th>
<th>Coil 2</th>
<th>Coil 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>a.1</td>
<td>on</td>
<td>off</td>
<td>off</td>
<td>Off</td>
</tr>
<tr>
<td>a.2</td>
<td>off</td>
<td>on</td>
<td>off</td>
<td>Off</td>
</tr>
<tr>
<td>a.3</td>
<td>off</td>
<td>off</td>
<td>on</td>
<td>Off</td>
</tr>
<tr>
<td>a.4</td>
<td>off</td>
<td>off</td>
<td>off</td>
<td>On</td>
</tr>
</tbody>
</table>

This sequence produces the smoothest movement and consumes least power.
B. Two-Coil Excitation - Each successive pair of adjacent coils is energized in turn.

<table>
<thead>
<tr>
<th>Step</th>
<th>Coil 4</th>
<th>Coil 3</th>
<th>Coil 2</th>
<th>Coil 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>b.1</td>
<td>on</td>
<td>on</td>
<td>off</td>
<td>Off</td>
</tr>
<tr>
<td>b.2</td>
<td>off</td>
<td>on</td>
<td>on</td>
<td>Off</td>
</tr>
<tr>
<td>b.3</td>
<td>off</td>
<td>off</td>
<td>on</td>
<td>On</td>
</tr>
<tr>
<td>b.4</td>
<td>on</td>
<td>off</td>
<td>off</td>
<td>On</td>
</tr>
</tbody>
</table>

This is not as smooth and uses more power but produces greater torque.

(If this is the sequence generated by the TM100 Disk Drive's Logic PCB, then how come the KP4M4-001 motor takes 100 steps per complete revolution?)

Note

The excitation of Coil 1 is always the inverse of the excitation of Coil 3. So, with the right circuit the excitation of Coil 4 is always the inverse of the excitation of Coil 2. You can generate this sequence with only two data lines. Interleaving the two sequences will cause the motor to half-step.
## Automated Unmanned Level Railway Crossing System

<table>
<thead>
<tr>
<th>Step</th>
<th>Coil 4</th>
<th>Coil 3</th>
<th>Coil 2</th>
<th>Coil 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>a.1</td>
<td>on</td>
<td>off</td>
<td>off</td>
<td>Off</td>
</tr>
<tr>
<td>b.1</td>
<td>on</td>
<td>on</td>
<td>off</td>
<td>Off</td>
</tr>
<tr>
<td>a.2</td>
<td>off</td>
<td>on</td>
<td>off</td>
<td>Off</td>
</tr>
<tr>
<td>b.2</td>
<td>off</td>
<td>on</td>
<td>on</td>
<td>Off</td>
</tr>
<tr>
<td>a.3</td>
<td>off</td>
<td>off</td>
<td>on</td>
<td>Off</td>
</tr>
<tr>
<td>b.3</td>
<td>off</td>
<td>off</td>
<td>on</td>
<td>On</td>
</tr>
<tr>
<td>a.4</td>
<td>off</td>
<td>off</td>
<td>off</td>
<td>On</td>
</tr>
<tr>
<td>b.4</td>
<td>on</td>
<td>off</td>
<td>off</td>
<td>On</td>
</tr>
</tbody>
</table>
CHAPTER 5
APPLICATIONS & ADVANTAGES
5.1 APPLICATIONS

- Real time transport systems.

5.2 ADVANTAGES

- Accident avoidance.
- Human Resource.
- Safety and quality of services.
CHAPTER 6
CONCLUSION
6. CONCLUSION

The accidents are avoided at places where there is no person manning the railway crossing gates.

Here we use the stepper motor to open and close the gates automatically when it is rotated clockwise or anticlockwise direction.

When the train arrives in a particular direction the transmitter IR senses and generates appropriate signal, then at the same time the receiver IR receives the signal and generates an interrupt.

When the interrupt is generated the stepper motor rotates in clockwise direction. When the interrupt ends the stepper motor rotates in anti clockwise direction.
SOURCE CODE
CODING

#include<at89x52.h>

/******************** Port Initialization
*************************************/

#define motor P2
#define ir1 P1_1
#define ir2 P1_0
#define red P3_4
#define yellow P3_5
#define green P3_6

/******************** generating a delay of 1 sec
****************************************************/

Delay ()
{
    int i;
    for (i=0;i<=2500;i++);
}

/******************** generating a delay of 10 sec
****************************************************/

delay1()
{
    int i;
    for (i=0;i<=25000;i++);
}

/******************** Rotating motor in clockwise direction
****************************************************/

motor_cw ()
{
    int j;
    for (j=0;j<4;j++)
    {
        P2=0x66;
        Delay ();
        P2=0x33;
    }
}

S.R.T.I.S.T. 83
Automated Unmanned Level Railway Crossing System

delay();
P2=0x99;
delay();
P2=0xcc;
delay();
}
}

/******************* Rotating motor in anti clockwise direction
***************************/

motor_acw()
{
    int j;
    for(j=0;j<4;j++)
    {
        P2=0x66;
delay();
P2=0xcc;
delay();
P2=0x99;
delay();
P2=0x33;
delay();
    }
}

/******************* Beginning of the main program
***************************/

main()
{
    unsigned int a=0, b=1; /* Initialising temporary variables */

    while(1)
    {
        red=1;       /* Red light off */
yellow=1;       /* Yellow light off */
green=0;       /* Green light on */

        if((ir1==0 || ir2==0) && a==0) /* Condition to check the arrival of the train */
        {
            motor_cw();   /* Closing the gate */
            red=0;
yellow=1;
green=1;
        }
    }
}
for(;((ir1==1 && ir2==0) || (ir1==0 && ir2==1));) /* Condition for the gate to be closed while the train is present at the two sensors */
{
    if((ir1==1 && ir2==1) || (ir1==1 || ir2==1) || (ir1==0 && ir2==0))
    {
        a=1;
        b=0;
    }
}

if((ir1==0 && ir2==1) || (ir2==0 && ir1==1)||(ir1==1 && ir2==1)) /* Condition for the gate to be closed while the train is in between the two sensors */
{
    for(;(ir1==1 && ir2==1);)
    {
        a=1;
        b=0;
    }
}

if((ir2==1 && ir1==1) && b==0) /* Condition to check the departure of train from the railway crossing */
{
    red=1;
    yellow=0;
    motor_acw(); /* Opening the gate */
    delay();
    delay1();
    yellow=1;
    green=0;
    delay();
    delay1();
    delay();
    delay();
    green=1;
    b=1;
    a=0;
}
BOOKS REFERRED


- Part of stepping motors by Douglas W.Jones, the university of IOWA Department of computer science.

WEBSITES

- www.8051.com
- www.8051microcontroller.com
- www.8051faq.com
- www.microchip.com
- www.etext.org