EMBEDDED SYSTEM BASED TRAIN COLLISION AVOIDANCE SYSTEM

A PROJECT REPORT

P- 2346

Submitted by

S ARULMANI	71204105005
V S MOHANKUMAR	71204105304

In partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

ELECTRICAL AND ELECTRONICS ENGINEERING

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE – 641 006

ANNA UNIVERSITY: CHENNAI 600 025

APRIL – 2008

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report entitled **"Embedded system based train Collision avoidance system"** is the bonafide work of

S . ARULMANI	**71204105005**
V. S .MOHANKUMAR	**71204105304**

Who carried out the project work under my supervision

[illegible]

SIGNATURE OF

THE HEAD OF THE DEPARTMENT

Prof. K. Regupathy Subramanian

[illegible]

SIGNATURE OF

THE GUIDE

Mr. V. Kandasamy

Certified that the candidate with university Register No. 71204105304 (AND) 71204105005 was examined in project viva voce Examination held on 19.4.2008

[illegible]

Internal Examiner

[illegible]

External Examiner

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY,

COIMBATORE – 641006.

ABSTRACT

ABSTRACT

In the railways safety and security are important factors. For safety the train accident occurs on unmanned gate crossing, train collision in the same track due to human mislead.
Each and every person's life is valuable to travel in the train journey. Our proposed design is to avoid train collision and unmanned gate control. This design is implemented using embedded system, IR transceivers, RF transmitter and receivers.

Now a days train accidents are occurring frequently in India. One of the main reasons for train accident is the traveling of two trains in same track in opposite direction. In order to avoid the accidents due to the above reason, we have designed this project.

This project is carried over in embedded system with the help of PIC microcontroller and RF transmitter and receiver. This project identifies the status of each train using IR transceivers and inform it to other trains and to the near by sub station. Each train will consist of RF transceiver which transmits its status and receives the status of other trains. RF communication is used to communicate; its range is more than sufficient to avoid this kind of accident.

This project is used to avoid the train collision. Hence we save the valuable human lives and losses. So this project is useful for railway department

ACKNOWLEDGEMENT

ACKNOWLEDEGEMENT

Our heartfelt thanks goes to Arutselvar **Dr. N. Mahalingam** B.sc. F.I.E.. Chairman. and our Vice Chairman Prof. **Dr. K. Arumugam** B.E. (Hons). M.S. (USA). M.I.E., of our college for having provided us with all necessary infrastructure and other facilities. to successfully carry out this project.

We thank our beloved Principal **Dr. Joseph V. Thanikal** M.E., Ph.D.. PDF. CEPIT. for providing all facilities to carry out this project work.

We express our deep sense of gratitude and thanks to **Prof. K. Regupathy Subramanian**. M.Sc., Dean and HOD of the Electrical and Electronics Engineering. for his valuable guidance with constant encouragement and motivation. which triggered us to a great extent in completing the project work successfully.

We would like to express our deep sense of gratitude and profound thanks to our guide **Mr.V. KANDASAMY,** M.Tech.. Lecturer. Electrical and Electronics Engineering Department, for his valuable guidance. support. constant encouragement and co-operation rendered throughout the project.

We are indebted to all Teaching and Non-Teaching members of the department of Electrical and Electronics Engineering and other departments for their timely help and valuable suggestions.

CONTENTS

CONTENTS

LIST OF FIGURES

INTRODUCTION

1.INTRODUCTION:

Now a days train accidents are occurring frequently due to unmanned railway crossing. collision of two trains on the same track by signal misleading by human. Such problems are reduced by designing the proper monitoring and control units in the railways.

Our design is to identify the train status on the track. collision avoidance and unmanned railway crossing gate control. This design is implemented by embedded system and wireless sensors and transceivers. The train status are identified by IR transceivers are placing on the either side of track with the distance of 5KM once. This IR transceiver is coupled with RF transmitters. The embedded processor is PIC16F877A with RF receivers. When a train is in A position that IR transceiver gives the signal to RF transmitter transmitting the signal to RF receiver coupled with PIC controller, the controller recognize that signal identifies place of train on the track.

A track consisting of IR transceivers in equal distances giving the train position a train enters on the track within the railway station we can identify the place of train on the track by embedded system at the same time another train enters in the opposite direction in the same track that train positions are monitored from the station. When the two trains are coming closer with the distance of 5Km our PIC controller automatically respond and transmit a signal to train power supply to switch off by a RF receiver. At the same time unmanned railway gate controlled by a embedded system by providing IR sensors on either side of gate. When a train enters towards gate the IR sensor giving the information to controller automatically the gate will close and train leave from gate the gate will open.

Here the embedded processor is PIC16F877A because it is a RISC type processor. 40 pin IC. inbuilt ADC's operating voltage of 5V DC with the speed of 4MHZ .

BLOCK DIAGRAM

2. BLOCK DIAGRAM:

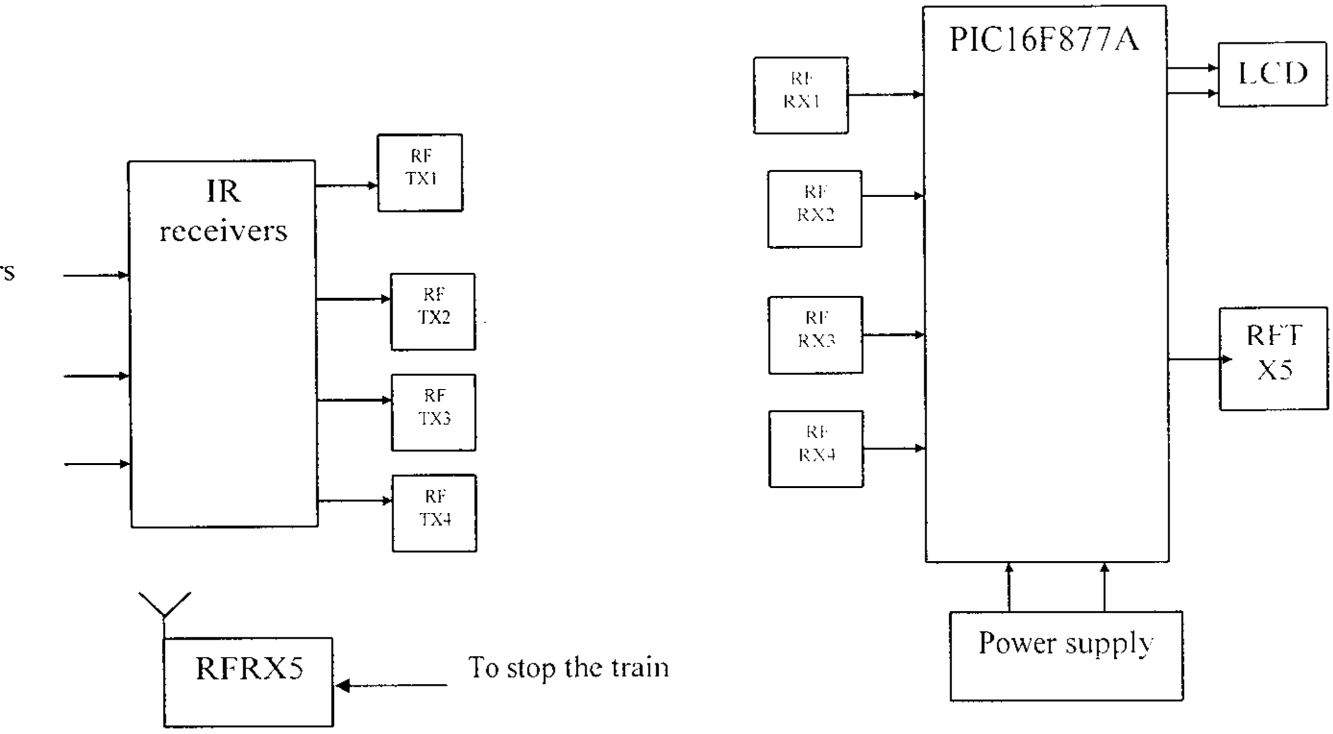


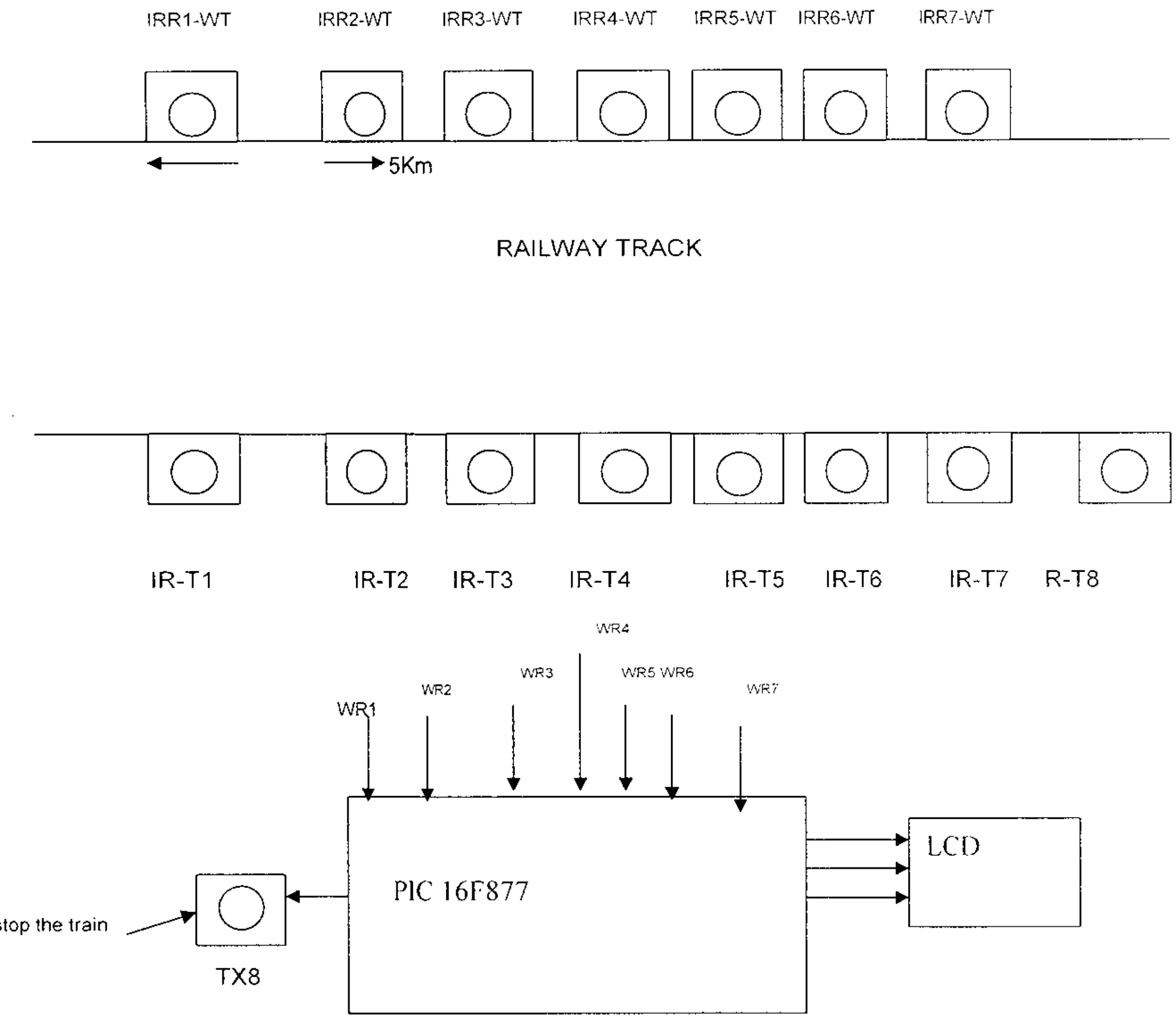
Fig.1.Block diagram of transmitter and receiver section

Block Diagram Description

Fig .1 shows the block diagram of transmitter and receiver section. In the transmitter section four IR transceivers are placed on either side of track at the distance of 5km.4 RF transmitter are connected to each. This design consists of 4 RF transceivers. Each RF transmitter connected to IR receivers. In receiver side a PIC16F877 controller coupled with 4 RF receivers. one RF transmitter to stop the train and LCD used to show the train condition.

LAYOUT BLOCK DIAGRAM

LAYOUT BLOCK DIAGRAM:



-R1 → IR Sensor of Receiver1
-R2 → IR Sensor of Receiver2

IR-T1→ IR Sensor of Transmitter1
IR-T1→ IR Sensor of Transmitter2

R – wireless receiver
T wireless transmitter

Fig.2. Layout diagram of components

.1. CIRCUIT DIAGRAM FOR TRAIN HUB:

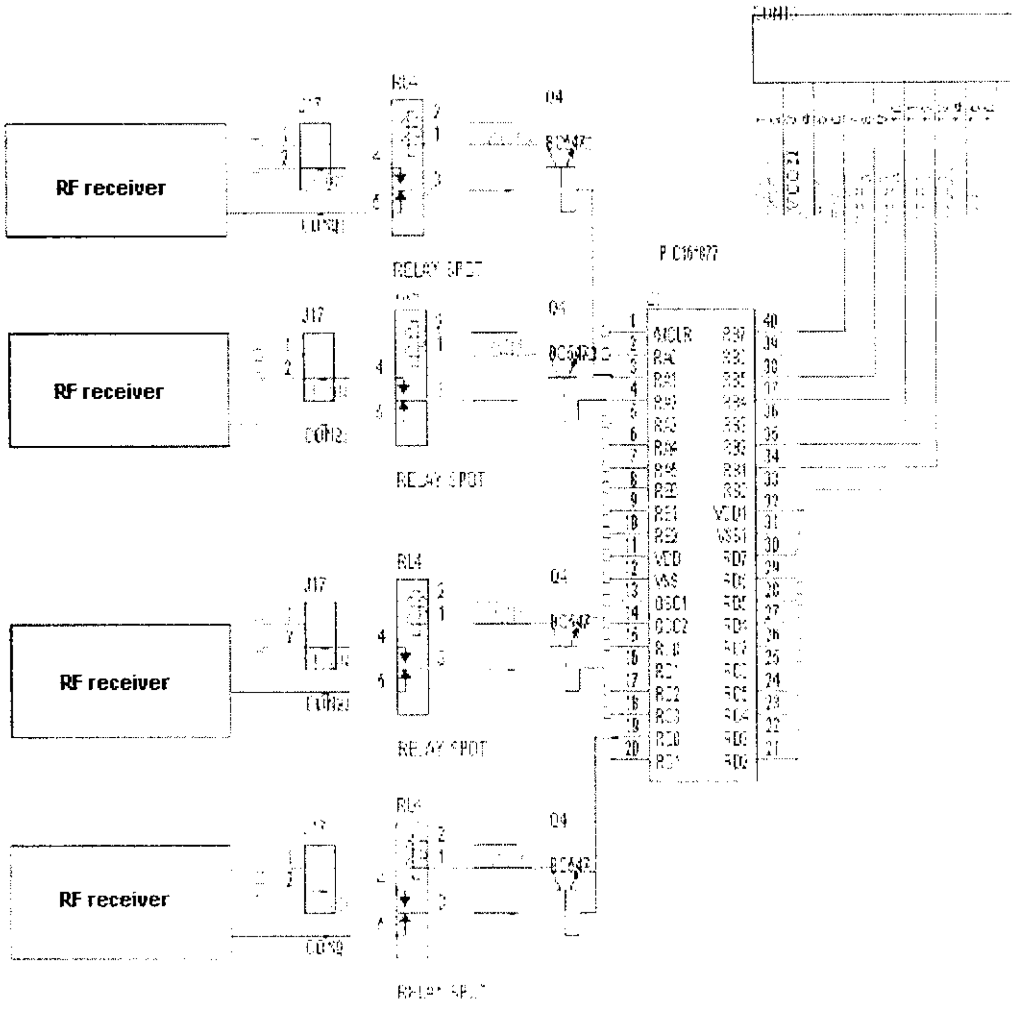


Fig.3. Circuit diagram of embedded system in the station

3.2. CIRCUIT DIAGRAM OF TRANSMITTER SECTION:

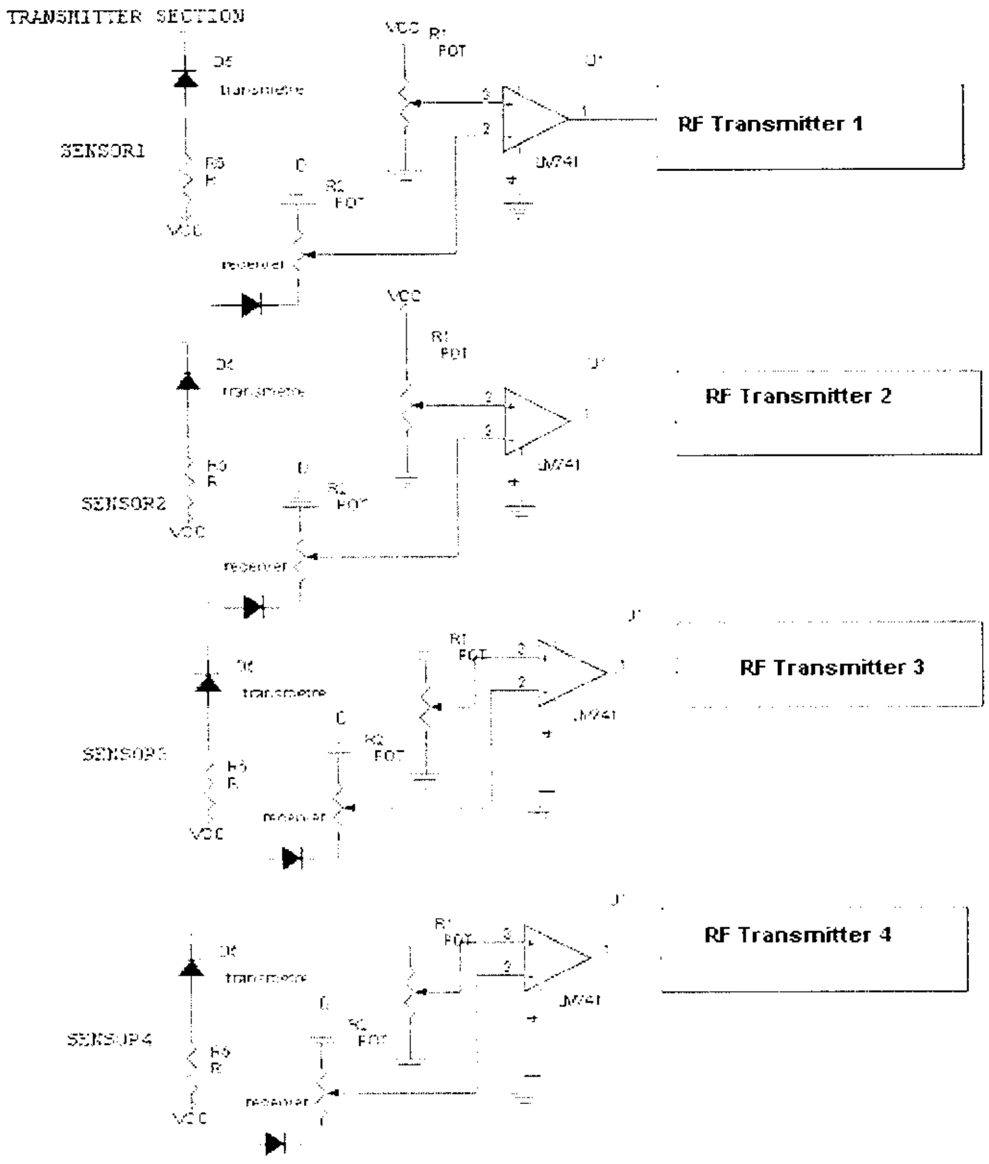


Fig.4. Circuit diagram of transmitter section

WORKING PRINCIPLE

4. WORKING PRINCIPLE:

In proposed control technique implemented with help of PIC16F877A microcontroller, infrared transmitter, infrared receiver, RF transmitter and receiver. The IR transmitter and receiver are placed on either side of track electric pole. The same arrangement implemented at every 5KM once. This IR receiver interfaced with RF transmitter this transmitter transmits the signal to control HUB placed in the nearest railway station. In the railway station PIC controller interfaced with the RF receivers with different frequency.

When a train enters the track the IR sensor detect the train and gives the signal to RF transmitter, the transmitter transmit the signal to RF receiver it is interfaced with the PIC controller. This controller interfaced with Liquid Crystal Display it is displaying the train position at the same time another train enters in the opposite side the corresponding sensors indicating the train position and place. When the controller detect the adjacent RF receiver signal there is a possibility of collision, immediately the controller gives the signal to RF transmitter and the track supply automatically switch off.

The same design incorporates the human less gate control. The infrared transmitter and receiver placed on the track at the distance of 4KM away from gate. When the train enters before 4KM of gate the receiver respond and automatically the gate closed and indicate red LED indication, when the train leaves from the gate the IR receiver gives the signal to controller and open the gate with help of proper mechanical arrangement.

P-2346

FLOW CHART

FLOW CHART:

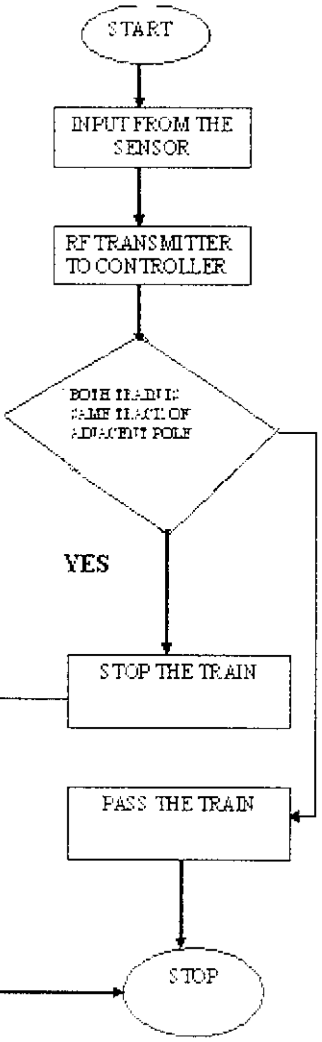

START
INPUT FROM THE SENSOR
RF TRANSMITTER TO CONTROLLER
BOTH TRAIN IS SAME TRACK OF ADJACENT POLE
YES
NO
STOP THE TRAIN
PASS THE TRAIN
STOP

Fig.5.flow chart for PIC program

The sensor signal from the train track is transmitted to the controller through RF transmitter in ascending or descending manner.

PIC microcontroller checks whether the signal is received from both the adjacent poles. if yes . stop the train else pass the train.

COMPONENT DETAILS

6. COMPONENT DETAILS:

6.1 PIC MICROCONTROLLER:

CORE FEATURES:

- High-performance RISC CPU
- Only 35 single word instructions to learn
- Operating speed: DC - 20 MHz clock input

  DC - 200 ns instruction cycle
- Up to 8K x 14 words of Flash Program Memory.

  Up to 368 x 8 bytes of Data Memory (RAM)

  Up to 256 x 8 bytes of EEPROM data memory
- Interrupt capability (up to 14 internal/external
- Eight level deep hardware stack
- Direct. indirect. and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC Oscillator for reliable operation
- Programmable code-protection
- Power saving SLEEP mode
- Selectable oscillator options
- In-Circuit Serial Programming (ICSP) via two pins
- Only single 5V source needed for programming capability
- In-Circuit Debugging via two pins
- Wide operating voltage range: 2.5V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial and Industrial temperature ranges
- Low-power consumption:

  2 mA typical @ 5V, 4 MHz

  20mA typical @ 3V. 32 kHz

  1mA typical standby current

PERIPHERAL FEATURES:

• Timer0: 8-bit timer/counter with 8-bit prescaler

• Timer1: 16-bit timer/counter with prescaler

• Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler

• Two Capture, Compare, PWM modules

Capture is 16-bit, max resolution is 12.5 ns.

Compare is 16-bit, max resolution is 200 ns,

PWM max. resolution is 10-bit

• 10-bit multi-channel Analog-to-Digital converter

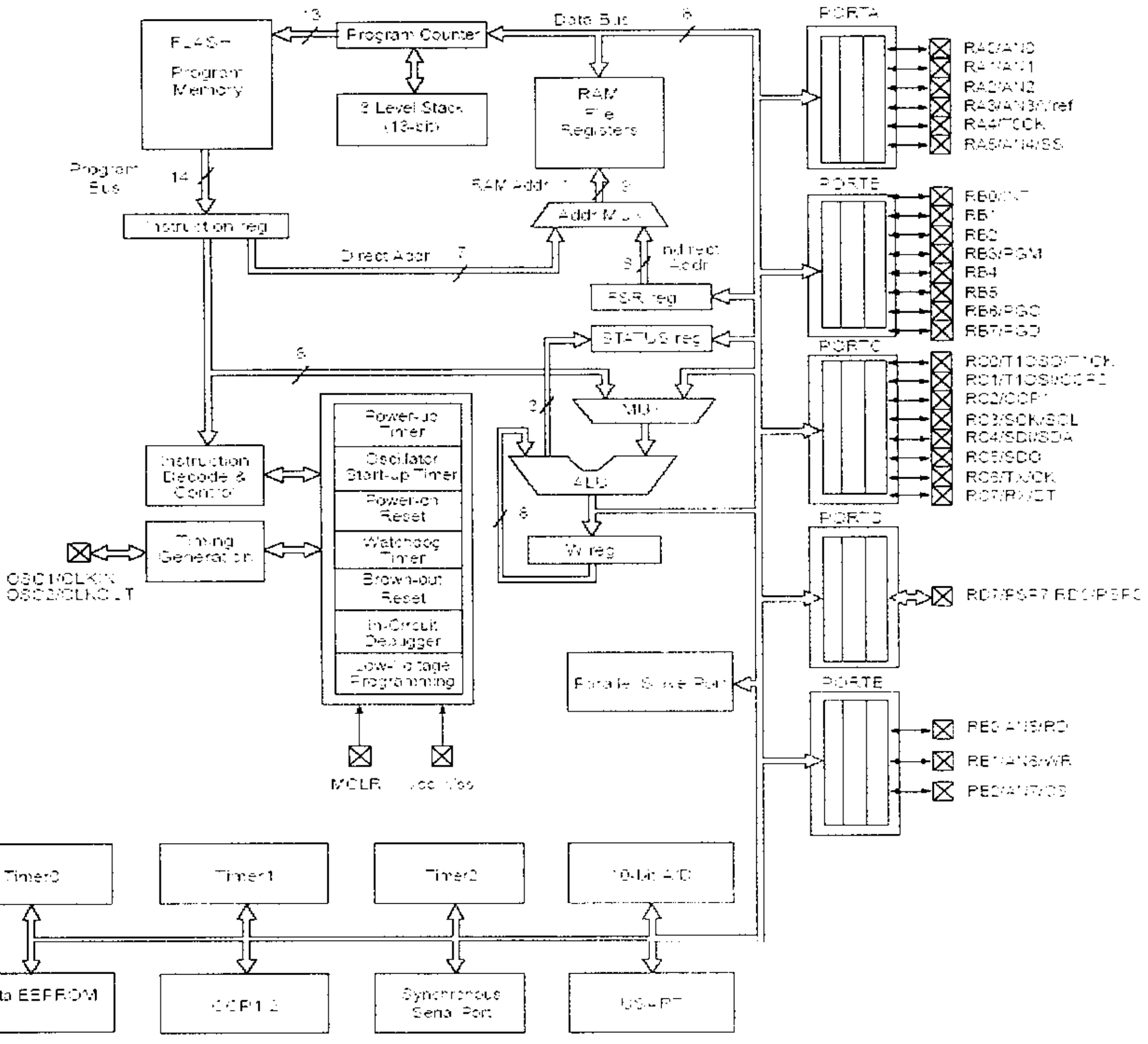
• Synchronous Serial Port (SSP) with SPI. (Master Mode) and I2C. (Master/Slave)

• USART/SCI with 9-bit address detection.

• Parallel Slave Port (PSP) 8-bits wide, with external RD, WR and CS controls

ARCHITECTURE OF PIC 16F877

Device	Program Flash	Data Memory	Data EEPROM
PIC16F874	4K	192 Bytes	128 Bytes
PIC16F877	8K	368 Bytes	256 Bytes



Note 1: Higher order bits are from the STATUS register

Fig.6. Architecture of PIC 16F877

16F877 Pin diagram

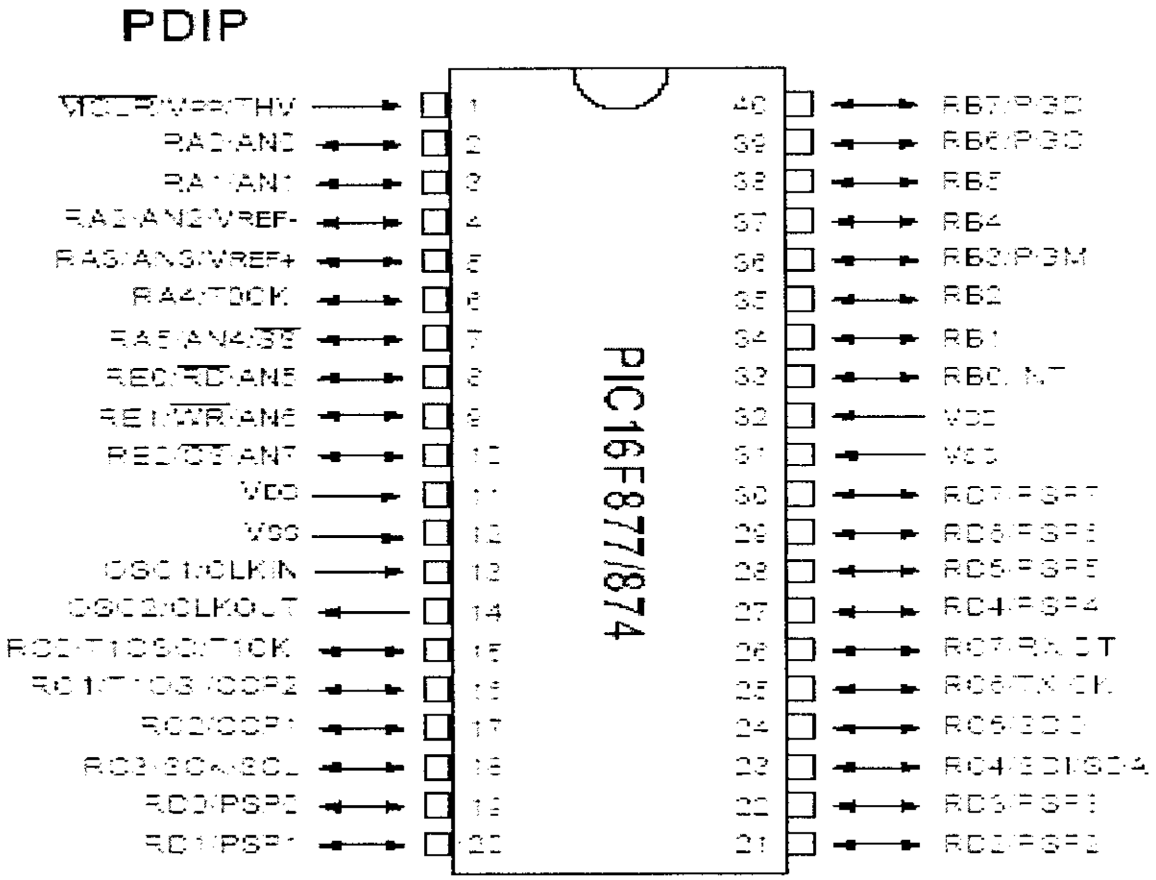


Fig.7.pin diagram of 16F877

I/O PORTS

Some pins for these I/O ports are multiplexed with an alternate function for the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

PORT A and the TRISA Register

PORT A is a 6-bit wide bi-directional port. The corresponding data direction register is TRISA. Setting a TRISA bit (=1) will make the corresponding PORT A pin an input, i.e., put the corresponding output driver in a Hi-impedance mode. Clearing a

TRISA bit (=0) will make the corresponding PORTA pin an output, i.e., put the contents of the output latch on the selected pin.

Reading the PORT A register reads the status of thepins whereas writing to it will write to the port latch. All write operations are read-modify-write operations. Therefore a write to a port implies that the port pins are read; this value is modified, and then written to the port data latch. Pin RA4 is multiplexed with the Timer0 module clock input to become the RA4/T0CKI pin. The RA4/T0CKI pin is a Schmitt Trigger input and an open drain output. All other RA port pins have TTL input levels and full CMOS output drivers. Other PORT A pins are multiplexed with analog inputs and analog VREF input. The operation of each pin is selected by clearing/setting the control bits in the ADCON1 register (A/D Control Register1).

The TRISA register controls the direction of the RA pins, even when they are being used as analog inputs. The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs.

PORT A Function

Name	Bit#	Buffer	Function
RA0/AN0	bit0	TTL	Input/output or analog input
RA1/AN1	bit1	TTL	Input/output or analog input
RA2/AN2	bit2	TTL	Input/output or analog input
RA3/AN3/VREF	bit3	TTL	Input/output or analog input or VREF
RA4/T0CKI	bit4	ST	Input/output or external clock input for Timer0 Output is open drain type
RA5/$\overline{SS}$/AN4	bit5	TTL	Input/output or slave select input for synchronous serial port or analog input

PORT B and the TRISB Register

PORT B is an 8-bit wide bi-directional port. The corresponding data direction register is TRISB. Setting a TRISB bit (=1) will make the corresponding PORT B pin an input, i.e., put the corresponding output driver in a hi-impedance mode.

Clearing a TRISB bit (=0) will make the corresponding PORT B pin an output. i.e., put the contents of the output latch on the selected pin. Three pins of PORT B are multiplexed with the Low Voltage Programming function: RB3/PGM, RB6/PGC and RB7/PGD. The alternate functions of these pins are described in the Special Features Section. Each of the PORT B pins has a weak internal pull-up. A single control bit can turn on all the pull-ups. This is performed by clearing bit RBPU (OPTION_REG<7>). The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset.

Four of PORT B's pins, RB7:RB4, have an interrupt on change feature. Only pins configured as inputs can cause this interrupt to occur (i.e. any RB7:RB4 pin configured as an output is excluded from the interrupt on change comparison). The input pins (of RB7:RB4) are compared with the old value latched on the last read of PORT B. The "mismatch" outputs of RB7:RB4 are OR'ed together to generate the RB Port Change Interrupt with flag bit RBIF (INTCON<0>). This interrupt can wake the device from SLEEP. The user, in the interrupt service routine, can clear the interrupt in the following manner:

a) Any read or write of PORT B. This will end the mismatch condition.

b) Clear flag bit RBIF. A mismatch condition will continue to set flag bit RBIF. Reading PORT B will end the mismatch condition, and allow flag bit RBIF to be cleared. The interrupt on change feature is recommended for wake-up on key depression operation and operations where PORT B is only used for the interrupt on change feature. Polling of PORT B is not recommended while using the interrupt on change feature. This interrupt on mismatch feature, together with software configurable pull-ups on these four pins, allow easy interface to a keypad and make it possible for wake-up on key depression

PORT B FUNCTIONS

Name	Bit#	Buffer	Function
RB0/INT	bit0	TTL/ST[1]	Input/output pin or external interrupt input. Internal software programmable weak pull-up.
RB1	bit1	TTL	Input/output pin. Internal software programmable weak pull-up.
RB2	bit2	TTL	Input/output pin. Internal software programmable weak pull-up.
RB3/PGM	bit3	TTL	Input/output pin or programming pin in LVP mode. Internal software programmable weak pull-up.
RB4	bit4	TTL	Input/output pin (with interrupt on change). Internal software programmable weak pull-up.
RB5	bit5	TTL	Input/output pin (with interrupt on change). Internal software programmable weak pull-up.
RB6/PGC	bit6	TTL/ST[2]	Input/output pin (with interrupt on change) or In-Circuit Debugger pin. Internal software programmable weak pull-up. Serial programming clock.
RB7/PGD	bit7	TTL/ST[2]	Input/output pin (with interrupt on change) or In-Circuit Debugger pin. Internal software programmable weak pull-up. Serial programming data.

Legend: TTL = TTL input, ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.

2: This buffer is a Schmitt Trigger input when used in serial programming mode.

PORT C and the TRISC Register

PORT C is an 8-bit wide bi-directional port. The corresponding data direction register is TRISC. Setting a TRISC bit (=1) will make the corresponding PORT C pin an input, i.e., put the corresponding output driver in a hi-impedance mode. Clearing a TRISC bit (=0) will make the corresponding PORT C pin an output, i.e., put the contents of the output latch on the selected pin. PORT C is multiplexed with several peripheral functions(Table-3.5). PORT C pins have Schmitt Trigger input buffers.

When the I2C module is enabled, the PORT C (3:4) pins can be configured with normal I2C levels or with SMBUS levels by using the CKE bit (SSPSTAT <6>).

When enabling peripheral functions, care should be taken in defining TRIS bits for each PORTC pin. Some peripherals override the TRIS bit to make a pin an output, while other peripherals override the TRIS bit to make a pin an input. Since the TRIS bit override is in effect while the peripheral is enabled. read-modify write instructions (BSF. BCF, XORWF) with TRISC as destination should be avoided. The user should refer to the corresponding peripheral section for the correct TRIS bit settings.

TABLE 3: PORT C FUNCTIONS

Name	Bit#	Buffer Type	Function
RC0/T1OSO/T1CKI	bit0	ST	Input/output port pin or Timer1 oscillator output/Timer1 clock input
RC1/T1OSI/CCP2	bit1	ST	Input/output port pin or Timer1 oscillator input or Capture2 input/ Compare2 output/PWM2 output
RC2/CCP1	bit2	ST	Input/output port pin or Capture1 input/Compare1 output/PWM1 output
RC3/SCK/SCL	bit3	ST	RC3 can also be the synchronous serial clock for both SPI and I²C modes.
RC4/SDI/SDA	bit4	ST	RC4 can also be the SPI Data In (SPI mode) or data I/O (I²C mode).
RC5/SDO	bit5	ST	Input/output port pin or Synchronous Serial Port data output
RC6/TX/CK	bit6	ST	Input/output port pin or USART Asynchronous Transmit or Synchronous Clock
RC7/RX/DT	bit7	ST	Input/output port pin or USART Asynchronous Receive or Synchronous Data

Legend: ST = Schmitt Trigger input

PORT D and TRISD Registers

This section is not applicable to the 28-pin devices. PORT D is an 8-bit port with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output.

PORT D can be configured as an 8-bit wide microprocessor Port (parallel slave port) by setting control bit PSPMODE (TRISE<4>). In this mode. the input buffers are TTL.

TABLE 5: PORT D FUNCTIONS

Name	Bit#	Buffer Type	Function
RD0/PSP0	bit0	ST/TTL[1]	Input/output port pin or parallel slave port bit0
RD1/PSP1	bit1	ST/TTL[1]	Input/output port pin or parallel slave port bit1
RD2/PSP2	bit2	ST/TTL[1]	Input/output port pin or parallel slave port bit2
RD3/PSP3	bit3	ST/TTL[1]	Input/output port pin or parallel slave port bit3
RD4/PSP4	bit4	ST/TTL[1]	Input/output port pin or parallel slave port bit4
RD5/PSP5	bit5	ST/TTL[1]	Input/output port pin or parallel slave port bit5
RD6/PSP6	bit6	ST/TTL[1]	Input/output port pin or parallel slave port bit6
RD7/PSP7	bit7	ST/TTL[1]	Input/output port pin or parallel slave port bit7

Legend: ST = Schmitt Trigger input TTL = TTL input
Note 1: Input buffers are Schmitt Triggers when in I/O mode and TTL buffer when in Parallel Slave Port Mode

PORT E and TRISE Register

PORT E has three pins RE0/RD/AN5, RE1/WR/AN6 and RE2/CS/AN7, which are individually configurable as inputs or outputs. These pins have Schmitt Trigger input buffers.

The PORT E pins become control inputs for the microprocessor port when bit PSPMODE (TRISE<4>) is set. In this mode, the user must make sure that the TRISE<2:0> bits are set (pins are configured as digital inputs). Ensure ADCON1 is configured for digital I/O. In this mode the input buffers are TTL.

PORTE pins are multiplexed with analog inputs. When selected as an analog input, these pins will read as '0's. TRISE controls the direction of the RE pins, even when they are being used as analog inputs. The user must make sure to keep the pins configured as inputs when using them as analog inputs.

Table 7: PORT E FUNCTIONS

Name	Bit#	Buffer Type	Function
RE0/$\overline{\text{RD}}$/AN5	bit0	ST/TTL(1)	Input/output port pin or read control input in parallel slave port mode or analog input: $\overline{\text{RD}}$ 1 = Not a read operation 0 = Read operation. Reads PORTD register (if chip selected)
RE1/$\overline{\text{WR}}$/AN6	bit1	ST/TTL(1)	Input/output port pin or write control input in parallel slave port mode or analog input: $\overline{\text{WR}}$ 1 =Not a write operation 0 =Write operation. Writes PORTD register (if chip selected)
RE2/$\overline{\text{CS}}$/AN7	bit2	ST/TTL(1)	Input/output port pin or chip select control input in parallel slave port mode or analog input: $\overline{\text{CS}}$ 1 = Device is not selected 0 = Device is selected

Legend: ST = Schmitt Trigger input TTL = TTL input
Note 1: Input buffers are Schmitt Triggers when in I/O mode and TTL buffers when in Parallel Slave Port Mode

MEMORY ORGANISATION

There are three memory blocks in each of the PIC16f877 MUCs. The program memory and Data Memory have separate buses so that concurrent access can occur.

PROGRAM MEMORY ORGANISATION

The PIC16f877 devices have a 13-bit program counter capable of addressing 8K *14 words of FLASH program memory. Accessing a location above the physically implemented address will cause a wraparound.

The RESET vector is at 0000h and the interrupt vector is at 0004h.

DATA MEMORY ORGANISQTION

The data memory is partitioned into multiple banks which contain the General Purpose Registers and the special functions Registers. Bits RP1 (STATUS<6) and RP0 (STATYUS<5>) are the bank selected bits.

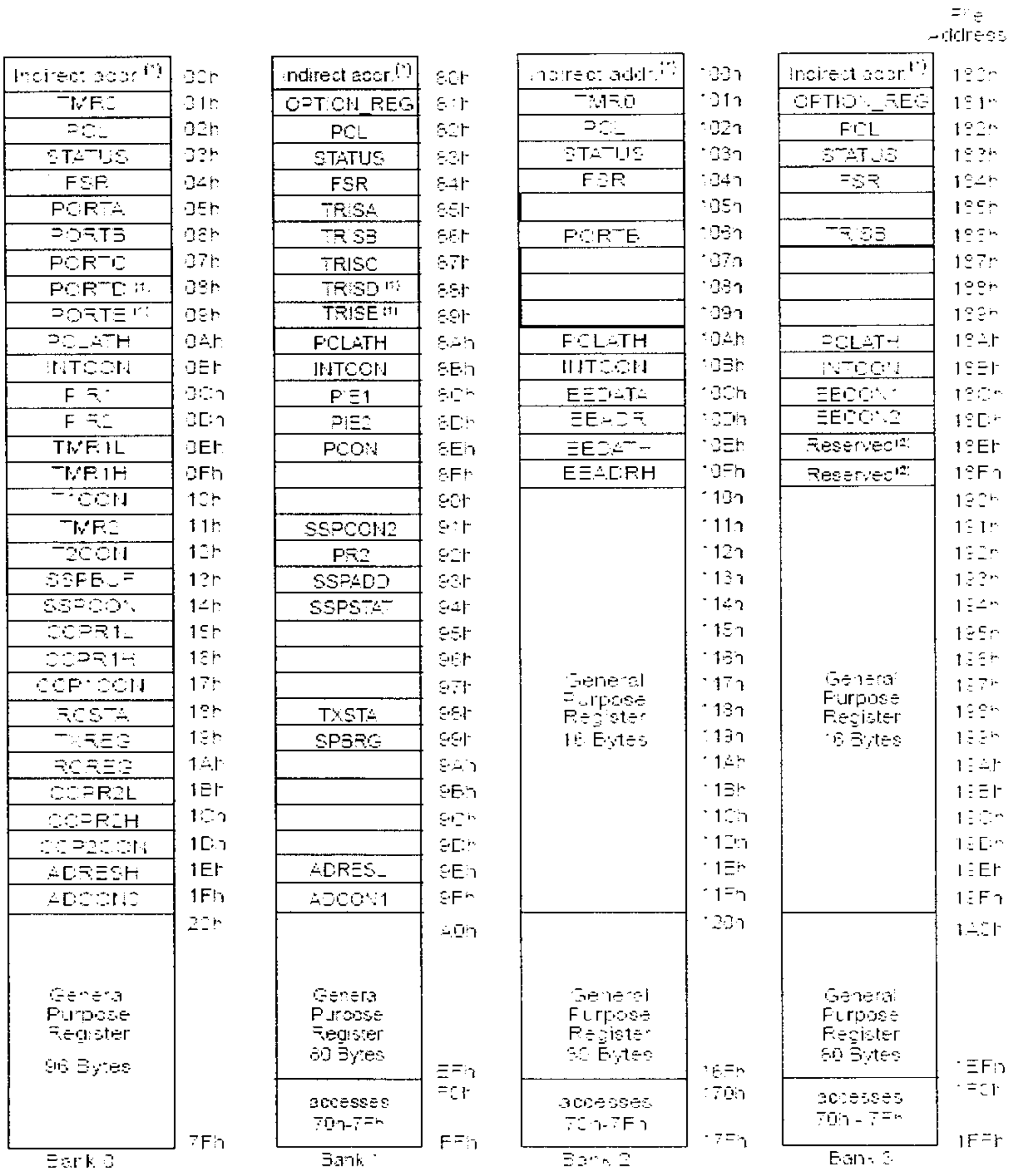
RP1:RP0	Banks
00	0
01	1
10	2
11	3

Each bank extends up to 7Fh (1238 bytes). The lower locations of each bank are reserved for the Special Function Registers. Above the Special Function Registers are General Purpose Registers, implemented as static RAM. All implemented banks contain special function registers. Some frequently used special function registers from one bank may be mirrored in another bank for code reduction and quicker access.

GENERAL PURPOSE REGISTER FILE

The register file can be accessed either directly or indirectly through the File Selected Register (FSR).

FIGURE (a): PIC16F877 REGISTER FILE MAP



File Address

Indirect addr.(*)	00h	Indirect addr.(*)	80h	Indirect addr.(*)	100h	Indirect addr.(*)	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
PORTD(1)	08h	TRISD(1)	88h		108h		188h
PORTE(1)	09h	TRISE(1)	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reserved(2)	18Eh
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reserved(2)	18Fh
T1CON	10h		90h	General Purpose Register 16 Bytes	110h	General Purpose Register 16 Bytes	190h
TMR2	11h	SSPCON2	91h		111h		191h
T2CON	12h	PR2	92h		112h		192h
SSPBUF	13h	SSPADD	93h		113h		193h
SSPCON	14h	SSPSTAT	94h		114h		194h
CCPR1L	15h		95h		115h		195h
CCPR1H	16h		96h		116h		196h
CCP1CON	17h		97h		117h		197h
RCSTA	18h	TXSTA	98h		118h		198h
TXREG	19h	SPBRG	99h		119h		199h
RCREG	1Ah		9Ah		11Ah		19Ah
CCPR2L	1Bh		9Bh		11Bh		19Bh
CCPR2H	1Ch		9Ch		11Ch		19Ch
CCP2CON	1Dh		9Dh		11Dh		19Dh
ADRESH	1Eh	ADRESL	9Eh		11Eh		19Eh
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
General Purpose Register 96 Bytes	20h	General Purpose Register 80 Bytes	A0h	General Purpose Register 80 Bytes	120h	General Purpose Register 80 Bytes	1A0h
			EFh		16Fh		1EFh
		accesses 70h-7Fh	F0h	accesses 70h-7Fh	170h	accesses 70h - 7Fh	1F0h
	7Fh		FFh		17Fh		1FFh
Bank 0		Bank 1		Bank 2		Bank 3	

— Unimplemented data memory locations, read as '0'.

* Not a physical register.

Note 1: These registers are not implemented on 28-pin devices.

2: These registers are reserved, maintain these registers clear.

Fig.8.16F877 register file map

INSTRUCTION SET SUMMARY

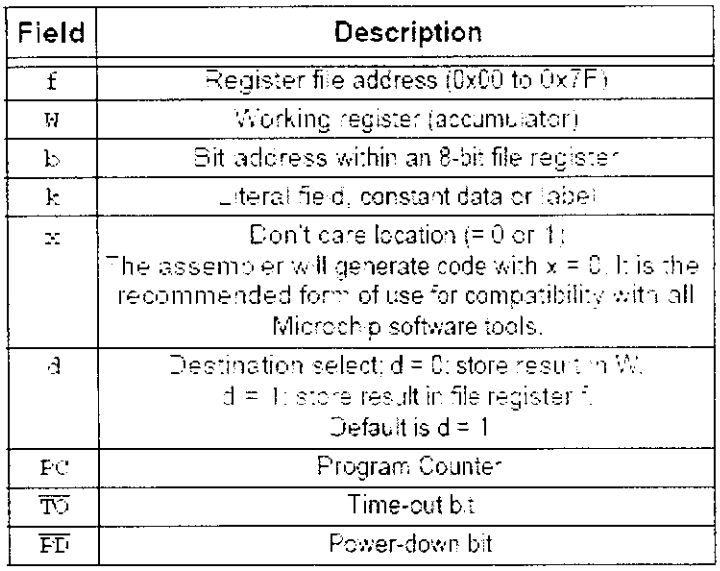
Each PIC 16f877 instruction is a 14-bit word, divided into an OPCODE which specifies the instruction type and one or more operand which further specify the operation of the instruction. The PIC16F877 instruction set summary in table 12 lists byte-oriented, bit-oriented, and literal and control operations. Table11 shows the opcode Field descriptions.

For byte-oriented instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator speciri4s which file register is to be used by the instruction. The destination designator specified where the result of the operation is to be placed. If'd' is zero, the result is placed in the w register. If'd' is one, the result is placed in the file register specified in the instruction.

For bit-oriented instructions, 'b' represents a bit field designator which selects the number of the bit affected by the operation, which 'f' represents the address of the file in which the bits is located.

For literal and control operations, 'k' represents an eight or eleven bit constant or literal value.

TABLE 9: OPCODE FIELD DESCRIPTIONS



Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1). The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1
PC	Program Counter
$\overline{TO}$	Time-out bit
$\overline{PD}$	Power-down bit

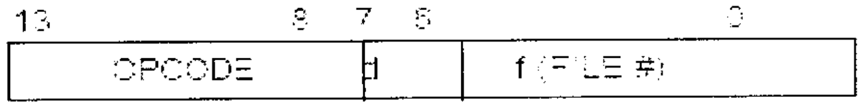
The instruction set is highly orthogonal and is grouped into three basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

All instructions are executed within one single instruction cycle, unless a conditional test is true or the program counter is changed as a result of an instruction. In this case, the execution takes two instruction cycles with the second cycle executed as a NOP. One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 ms. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 2 ms.

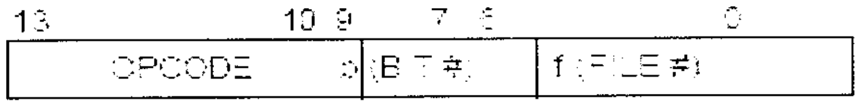
FIGURE (b): GENERAL FORMAT FOR INSTRUCTIONS

Byte-oriented file register operations



d = 0 for destination W
d = 1 for destination f
f = 7-bit file register address

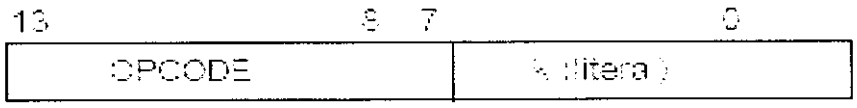
Bit-oriented file register operations



b = 3-bit bit address
f = 7-bit file register address

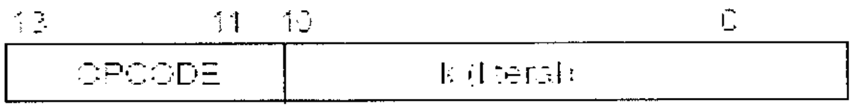
Literal and control operations

General



k = 8-bit immediate value

CALL and GOTO instructions only



k = 11-bit immediate value

16F877 INSTRUCTION SET

Mnemonic,	Operands	Description	Cycles	14-Bit Opcode				Status Affected	Notes
				MSb			LSb		
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C DC Z	1,2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	1,2
CLRF	f	Clear f	1	00	0001	1fff	ffff	Z	2
CLRW	-	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1,2,3
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	f	Move W to f	1	00	0000	1fff	ffff		
NOP	-	No Operation	1	00	0000	0xx0	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C DC Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1(2)	01	10bb	bfff	ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1(2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C DC Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	-	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}$ $\overline{PD}$	
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	-	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	-	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	-	Go into standby mode	1	00	0000	0110	0011	$\overline{TO}$ $\overline{PD}$	
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C DC Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

Note 1: When an I/O register is modified as a function of itself (e.g., MOVF PORTB, 1), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 Module.

3: If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

DATA EEPROM AND FLASH PROGRAM MEMORY

The Data EEPROM and FLASH Program Memory are readable and writable during normal operation over the entire VDD range. These operations take place on a single byte for Data EEPROM memory and a single word for Program memory. A write operation causes an erase-then-write operation to take place on the specified byte or word. A bulk erase operation may not be issued from user code (which includes removing code protection).

Access to program memory allows for checksum calculation. The values written to program memory do not need to be valid instructions. Therefore, up to 14-bit numbers can be stored in memory for use as calibration parameters, serial numbers, packed 7-bit ASCII, etc. Executing a program memory location containing data that form an invalid instruction, results in the execution
of a NOP instruction.

The EEPROM Data memory is rated for high erase/ writes cycle's .The FLASH program memory is rated much lower, because EEPROM data memory can be used to store frequently updated values. An on-chip timer controls the write time and it will vary with voltage and temperature, as well as from chip to chip. Please refer to the specifications for exact limits

A byte or word write automatically erases the location and writes the new value (erase before write). Writing to EEPROM data memory does not impact the operation of the device. Writing to program memory will cease the execution of instructions until the write is complete. The program memory cannot be accessed
during the write. During the write operation, the oscillator continues to run, the peripherals continue to function and interrupt events will be detected and essentially "queued" until the write is complete. When the write completes, the next instruction in the pipeline is executed and the branch to the interrupt vector will take place, if the interrupt is enabled and occurred during the write.

Read and write access to both memories take place indirectly through a set of Special Function Registers (SFR). The six SFRs used are:

• EEDATA

• EEDATH

• EEADR

• EEADRH

• EECON1

• EECON2

The EEPROM data memory allows byte read and writes operations without interfering with the normal operation of the microcontroller. When interfacing to EEPROM data memory, the EEADR register holds the address to be accessed. Depending on the operation, the EEDATA register holds the data to be written, or the data read, at the address in EEADR. The PIC16F873/874 devices have 128 bytes of EEPROM data memory and therefore, require that the MSB of EEADR remain clear. The EEPROM data memory on these devices does not wrap around to 0, i.e., 0x80 in the EEADR does not map to 0x00. The PIC16F876/877 devices have 256 bytes of EEPROM data memory and therefore, use all 8-bits of the EEADR.

The FLASH program memory allows non-intrusive read access, but writes operations cause the device to stop executing instructions, until the write completes. When interfacing to the program memory, the EEADRH: EEADR registers form a two-byte word, which holds the 13-bit address of the memory location being accessed. The register combination of EEDATH: EEDATA holds the 14-bit data for writes, or reflects the value of program memory after a read operation. Just as in EEPROM data memory accesses, the value of the EEADRH: EEADR registers must be within the valid range of program memory, depending on the device: 0000h to 1FFFh for the PIC16F873/874, or 0000h to 3FFFh for the PIC16F876/877. Addresses outside of this range do not wrap around to 0000h.

EECON1 and EECON2 Registers

The EECON1 register is the control register for configuring and initiating the access. The EECON2 register is not a physically implemented register, but is used exclusively in the memory write sequence to prevent inadvertent writes.

There are many bits used to control the read and write operations to EEPROM data and FLASH program memory. The EEPGD bit determines if the access will

be a program or data memory access. When clear, any subsequent operations will work on the EEPROM data memory. When set, all subsequent operations will
operate in the program memory.

Read operations only use one additional bit, RD, which initiates the read operation from the desired memory location. Once this bit is set, the value of the desired memory location will be available in the data registers. This bit cannot be cleared by firmware. It is automatically cleared at the end of the read operation. For EEPROM data memory reads, the data will be available in the EEDATA register in the very next instruction cycle after the RD bit is set. For program memory reads, the data will be loaded into the EEDATH: EEDATA registers; following the second instruction after the RD bit is set.

Write operations have two control bits, WR and WREN, and two status bits, WRERR and EEIF. The WREN bit is used to enable or disable the write operation. When WREN is clear, the write operation will be disabled. Therefore, the WREN bit must be set before executing a write operation. The WR bit is used to initiate the write operation. It also is automatically cleared at the end of the write operation. The interrupt flag EEIF is used to determine when the memory write completes. This flag must be cleared in software before setting the WR bit.
For EEPROM data memory, once the WREN bit and the WR bit have been set, the desired memory address in EEADR will be erased, followed by a write of the data in EEDATA. This operation takes place in parallel with the microcontroller continuing to execute normally. When the write is complete, the EEIF flag bit will be set.

For program memory, once the WREN bit and the WR bit have been set, the microcontroller will cease to execute instructions. The desired memory location pointed to by EEADRH: EEADR will be erased. Then, the data value in EEDATH: EEDATA will be programmed. When complete, the EEIF flag bit will be set and the microcontroller will continue to execute code. The WRERR bit is used to indicate when the PIC16F87X device has been reset during a write operation.
WRERR should be cleared after Power-on Reset. Thereafter, it should be checked on any other RESET. The WRERR bit is set when a write operation

is interrupted by a MCLR Reset, or a WDT Time-out Reset, during normal operation. In these situations, following a RESET, the user should check the WRERR bit and rewrite the memory location, if set. The contents of the data registers, address registers and EEPGD bit are not affected by either MCLR Reset, or WDT Timeout Reset, during normal operation.

EECON1 REGISTER (ADDRESS 18Ch)

R/W-x	U-0	U-0	U-0	R/W-x	R/W-0	R/S-0	R/S-0
EEPGD	—	—	—	WRERR	WREN	WR	RD
bit 7							bit 0

bit 7 EEPGD: Program/Data EEPROM Select bit

1 = Accesses program memory

0 = Accesses data memory

(This bit cannot be changed while a read or write operation is in progress)

bit 6-4 Unimplemented: Read as '0'

bit 3 WRERR: EEPROM Error Flag bit

1 = A write operation is prematurely terminated

(any MCLR Reset or any WDT Reset during normal operation)

0 = The write operation completed

bit 2 WREN: EEPROM Write Enable bit

1 = Allows write cycles

0 = Inhibits write to the EEPROM

bit 1 WR: Write Control bit

1 = Initiates a write cycle. (The bit is cleared by hardware once write is complete. The WR bit can only be set (not cleared) in software.)

0 = Write cycle to the EEPROM is complete

bit 0 RD: Read Control bit

1 = Initiates an EEPROM read. (RD is cleared in hardware. The RD bit can only be set (not cleared) in software.)

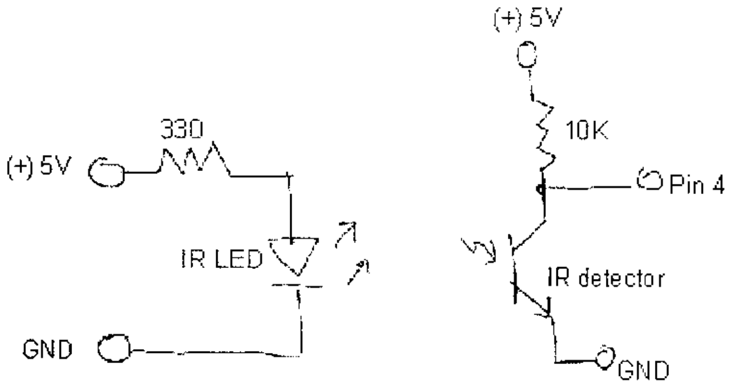
0 = Does not initiate an EEPROM read

6.2. IR TRANSMITTER RECEIVER:

Use an IR LED and phototransistor pair to create a light beam switch. Point the components at each other to turn the switch on, and then break the beam to turn the switch off. Use to detect when of your machine passes by a particular point. Or, bounce the light from the diode off a part to reflect back onto the detector. If the part is there, light will reach the detector and the signal can be passed to your Stamp. An IR LED/detector pair is exactly how your TV remote works. You can control your TV from across the room because the diode is pulsed briefly at a much higher current which gives off much more light.

IR Light Beam Circuits

Use an IR LED and phototransistor pair to create a light beam switch. Point the components at each other to turn the switch on, and then break the beam to turn the switch off. Use to detect when of your machine passes by a particular point. Or, bounce the light from the diode off a part to reflect back onto the detector. If the part is there, light will reach the detector and the signal can be passed to your Stamp. An IR LED/detector pair is exactly how your TV remote works. You can control your TV from across the room because the diode is pulsed briefly at a much higher current which gives off much more light.



IR emitter/detector hookup

IR LED: Connect like an ordinary LED using a 330 ohm series resistor to the +5 supply (or to a Stamp pin if you want to switch the source on and off). Current draw is about 11 mA with a 330 ohm resistor. Current runs from anode to cathode. Flat on the case marks the cathode.

IR Phototransistor: A phototransistor is just like a regular transistor except the base lead is disabled or absent and light activates base current. The flat on the case marks the collector, the other lead is the emitter. Connect the collector to one end of a 10K resistor and connect the other end of the resistor to a +5V supply (you can use the +5 pin on the Stamp). Connect the emitter to ground. To view output, measure the voltage at the collector with your DVM. The voltage should start out at +5V. When pointing the IR diode at the phototransistor, the voltage should drop down to near zero. To interface with the Stamp, make a second connection from the collector to a Stamp pin (for the example below use Pin 4).

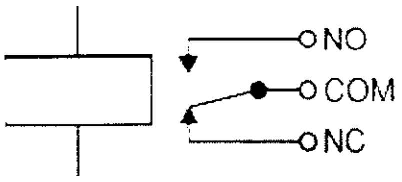
Telling them apart The IR LED and the IR phototransistor look alike. If you don't know which is which, you can try them in the test circuit described above, and swap if nothing happens. If you have the IR pair, look carefully into the clear plastic section. The gap between the two metal leads on the LED is smaller than the gap on the phototransistor. Also, the larger internal lead on the LED sticks across like a flag with a tiny bowl embedded on top while the phototransistor lead slants up with the tiny square chip glued to the top.

This is the simplest program you can use to test the operation of your IR sensor. Watch the numbers on the screen change between 1 and 0 as you block and unblock the sensor. If they don't change, measure the voltage at pin 4 while you block and unblock the sensor. If the voltage does not change, disconnect the wire going to pin 4 and measure the voltage on the wire (not on the pin). If it doesn't change, there is a wiring error or the sensor or emitter is bad. Note that the IRLED and phototransistor should be no more than two or three inches apart for this test.

Wire an ordinary red LED in series with a 330 ohm resistor to Pin 0. Wire the IR diode and phototransistor as described above. Hold the diode 0.25 to 3 inches from the phototransistor. The LED should light. Put your hand or a card between the diode and phototransistor and watch the LED turn off. For a slightly longer distance, drive the LED with more current. (A 100 ohm resistor will result in 38 mA, but be sure to hook it up to a separate +5 supply rather than a Stamp pin. Or, calculate the appropriate resistor and connect to a +12 supply.) Circuit is sensitive to ambient lighting. Can put each component in a dark tube to mask. Warning, if the circuit works perfectly in your room, it may or may not work in another location because the room lighting is different.

6.3. RELAY:

A relay is an **electrically operated switch**. Current flowing through the coil of the relay creates a magnetic field which attracts a lever and changes the switch contacts. The coil current can be on or off so relays have two switch positions and they are **double throw (changeover)** switches.



Relays allow one circuit to switch a second circuit which can be completely separate from the first. For example a low voltage battery circuit can use a relay to switch a 230V AC mains circuit. There is no electrical connection inside the relay between the two circuits, the link is magnetic and mechanical.

The coil of a relay passes a relatively large current, typically 30mA for a 12V relay, but it can be as much as 100mA for relays designed to operate from lower voltages. Most ICs (chips) cannot provide this current and a transistor is usually used to amplify the small IC current to the larger value required for the relay coil. The maximum output current for the popular 555 timer IC is 200mA so these devices can supply relay coils directly without amplification.

Relays are usually SPDT or DPDT but they can have many more sets of switch contacts, for example relays with 4 sets of changeover contacts are readily available. For further information about switch contacts and the terms used to describe them please see the page on switches.

Most relays are designed for PCB mounting but you can solder wires directly to the pins providing you take care to avoid melting the plastic case of the relay.

The supplier's catalogue should show you the relay's connections. The coil will be obvious and it may be connected either way round. Relay coils produce brief high voltage 'spikes' when they are switched off and this can destroy transistors and ICs in the circuit. To prevent damage you must connect a protection diode across the relay coil.

The animated picture shows a working relay with its coil and switch contacts. You can see a lever on the left being attracted by magnetism when the coil is switched on. This lever moves the switch contacts. There is one set of contacts (SPDT) in the foreground and another behind them, making the relay DPDT.

The relay's switch connections are usually labeled COM, NC and NO:

- **COM** = Common, always connect to this; it is the moving part of the switch.
- **NC** = Normally Closed, COM is connected to this when the relay coil is **off**.

- **NO** = Normally Open. COM is connected to this when the relay coil is **on**.
- Connect to COM and NO if you want the switched circuit to be **on when the relay coil is on.**
- Connect to COM and NC if you want the switched circuit to be **on when the relay coil is off**.

CHOOSING A RELAY

You need to consider several features when choosing a relay:

1. **Physical size and pin arrangement**
   If you are choosing a relay for an existing PCB you will need to ensure that its dimensions and pin arrangement are suitable. You should find this information in the supplier's catalogue.
2. **Coil voltage**
   The relay's coil voltage rating and resistance must suit the circuit powering the relay coil. Many relays have a coil rated for a 12V supply but 5V and 24V relays are also readily available. Some relays operate perfectly well with a supply voltage which is a little lower than their rated value.
3. **Coil resistance**
   The circuit must be able to supply the current required by the relay coil. You can use Ohm's law to calculate the current:

$$\text{Relay coil current} = \frac{\text{supply voltage}}{\text{coil resistance}}$$

4. For example: A 12V supply relay with a coil resistance of 400 passes a current of 30mA. This is OK for a 555 timer IC (maximum output current 200mA), but it is too much for most ICs and they will require a transistor to amplify the current.
5. **Switch ratings (voltage and current)**
   The relay's switch contacts must be suitable for the circuit they are to control. You

will need to check the voltage and current ratings. Note that the voltage rating is usually higher for AC, for example: "5A at 24V DC or 125V AC".

6. **Switch contact arrangement (SPDT, DPDT etc)**
   Most relays are SPDT or DPDT which are often described as "single pole changeover" (SPCO) or "double pole changeover" (DPCO). For further information please see the page on switches.

PROTECTION DIODES FOR RELAYS

Transistors and ICs (chips) must be protected from the brief high voltage 'spike' produced when the relay coil is switched off. The diagram shows how a signal diode (e.g. 1N4148) is connected across the relay coil to provide this protection. Note that the diode is connected 'backwards' so that it will normally **not** conduct. Conduction only occurs when the relay coil is switched off, at this moment current tries to continue flowing through the coil and it is harmlessly diverted through the diode. Without the diode no current could flow and the coil would produce a damaging high voltage 'spike' in its attempt to keep the current flowing.

ADVANTAGES OF RELAYS:

- Relays can switch **AC and DC**, transistors can only switch DC.
- Relays can switch **high voltages**, transistors cannot.
- Relays are a better choice for switching **large currents** (> 5A).
- Relays can switch **many contacts** at once.

DISADVANTAGES OF RELAYS:

- Relays are **bulkier** than transistors for switching small currents.
- Relays **cannot switch rapidly** (except reed relays), transistors can switch many times per second.
- Relays **use more power** due to the current flowing through their coil.

- Relays **require more current than many chips can provide**, so a low power transistor may be needed to switch the current for the relay's coil.

6.4. POWER SUPPLIES:

The present chapter introduces the operation of power supply circuits built using filters, rectifiers, and then voltage regulators. Starting with an ac voltage, a steady dc voltage is obtained by rectifying the ac voltage, then filtering to a dc level, and finally, regulating to obtain a desired fixed dc voltage. The regulation is usually obtained from an IC voltage regulator unit, which takes a dc voltage and provides a somewhat lower dc voltage, which remains the same even if the input dc voltage varies, or the output load connected to the dc voltage changes.

A block diagram containing the parts of a typical power supply and the voltage at various points in the unit is shown in fig 19.1. The ac voltage, typically 120 V rms, is connected to a transformer, which steps that ac voltage down to the level for the desired dc output. A diode rectifier then provides a full-wave rectified voltage that is initially filtered by a simple capacitor filter to produce a dc voltage. This resulting dc voltage usually has some ripple or ac voltage variation. A regulator circuit can use this dc input to provide a dc voltage that not only has much less ripple voltage but also remains the same dc value even if the input dc voltage varies somewhat, or the load connected to the output dc voltage changes. This voltage regulation is usually obtained using one of a number of popular voltage regulator IC units.

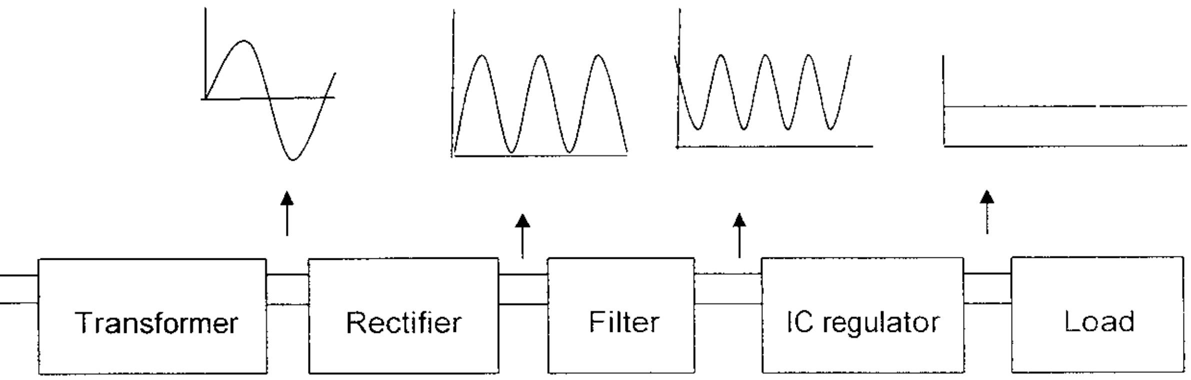


Fig.9. Power supply unit

IC VOLTAGE REGULATORS:

Voltage regulators comprise a class of widely used ICs. Regulator IC units contain the circuitry for reference source, comparator amplifier, control device, and overload protection all in a single IC. Although the internal construction of the IC is somewhat different from that described for discrete voltage regulator circuits, the external operation is much the same. IC units provide regulation of either a fixed positive voltage, a fixed negative voltage, or an adjustably set voltage.

A power supply can be built using a transformer connected to the ac supply line to step the ac voltage to desired amplitude, then rectifying that ac voltage, filtering with a capacitor and RC filter, if desired, and finally regulating the dc voltage using an IC regulator. The regulators can be selected for operation with load currents from hundreds of milli amperes to tens of amperes, corresponding to power ratings from milliwatts to tens of watts.

6.5. LCD DISPLAY:

Liquid crystal displays (LCDs) have materials which combine the properties of both liquids and crystals. Rather than having a melting point, they have a temperature range within which the molecules are almost as mobile as they would be in a liquid, but are grouped together in an ordered form similar to a crystal.

An LCD consists of two glass panels, with the liquid crystal material sand witched in between them. The inner surface of the glass plates are coated with transparent electrodes which define the character, symbols or patterns to be displayed polymeric layers are present in between the electrodes and the liquid crystal, which makes the liquid crystal molecules to maintain a defined orientation angle.

One each polarisers are pasted outside the two glass panels. These polarisers would rotate the light rays passing through them to a definite angle, in a particular direction

When the LCD is in the off state, light rays are rotated by the two polarisers and the liquid crystal, such that the light rays come out of the LCD without any orientation, and hence the LCD appears transparent.

When sufficient voltage is applied to the electrodes. the liquid crystal molecules would be aligned in a specific direction. The light rays passing through the LCD would be rotated by the polarizer, which would result in activating / highlighting the desired characters.

The LCD's are lightweight with only a few millimeters thickness. Since the LCD's consume less power, they are compatible with low power electronic circuits. and can be powered for long durations.

The LCD's do not generate light and so light is needed to read the display. By using backlighting, reading is possible in the dark. The LCD's have long life and a wide operating temperature range.

Changing the display size or the layout size is relatively simple which makes the LCD's more customer friendly.

The LCDs used exclusively in watches. calculators and measuring instruments are the simple seven-segment displays, having a limited amount of numeric data. The recent advances in technology have resulted in better legibility. more information displaying capability and a wider temperature range. These have resulted in the LCDs being extensively used in telecommunications and entertainment electronics. The LCDs have even started replacing the cathode ray tubes (CRTs) used for the display of text and graphics. and also in small TV applications.

RCM2034R

The RCM2034R is a reflective TN type liquid crystal module with a built-in controller / driver LSI and a display capacity of 16 characters _ 1 line.

Applications

Personal computers. word processors. facsimiles. telephones. etc.

Features

1) Wide viewing angle and high contrast.

2) 5_7 dot character matrix with cursor.

3) Interfaces with 4-bit or 8-bit MPUs.

4) Displays up to 226 characters and special symbols.

5) Custom character patterns are displayed with the character RAM.

6) Abundant instruction set including clear display, cursor on /off, and character blinking.

7) Compact and light weight for easy assembly to the host instrument.

8) Operable on single 5 V power supply.

9) Low power consumption.

BUZZER

7. BUZZER:

A **buzzer** or **beeper** is a signalling device, usually electronic, typically used in automobiles, household appliances such as a microwave oven, or game shows.

It most commonly consists of a number of switches or sensors connected to a control unit that determines if and which button was pushed or a preset time has lapsed, and usually illuminates a light on the appropriate button or control panel, and sounds a warning in the form of a continuous or intermittent buzzing or beeping sound. Initially this device was based on an electromechanical system which was identical to an electric bell without the metal gong (which makes the ringing noise). Often these units were anchored to a wall or ceiling and used the ceiling or wall as a sounding board. Another implementation with some AC-connected devices was to implement a circuit to make the AC current into a noise loud enough to drive a loudspeaker and hook this circuit up to a cheap 8-ohm speaker. Nowadays, it is more popular to use a ceramic-based piezoelectric sounder like a Sonalert which makes a high-pitched tone. Usually these were hooked up to "driver" circuits which varied the pitch of the sound or pulsed the sound on and off.

In game shows it is also known as a "lockout system," because when one person signals ("buzzes in"), all others are locked out from signalling. Several game shows have large buzzer buttons which are identified as "plungers".

The word "buzzer" comes from the rasping noise that buzzers made when they were electromechanical devices, operated from stepped-down AC line voltage at 50 or 60 cycles. Other sounds commonly used to indicate that a button has been pressed are a ring or a beep.

ELECTRIC MOTOR

8. ELECTRIC MOTOR:

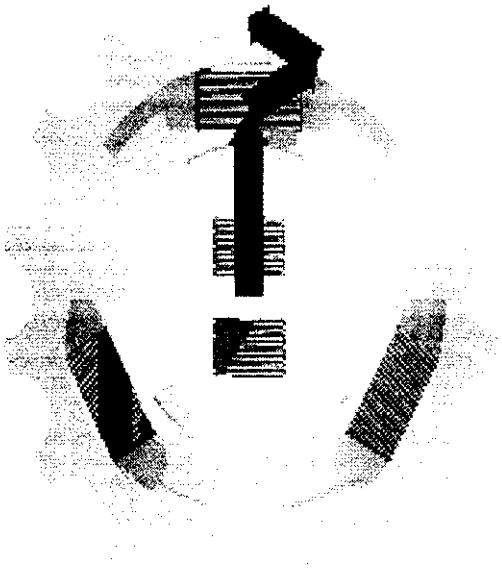


fig.10.Electric motor

Rotating magnetic field as a sum of magnetic vectors from 3 phase coils

An **electric motor** converts electrical energy into mechanical energy. The reverse task, that of converting mechanical energy into electrical energy, is accomplished by a generator or dynamo. Traction motors used on locomotives often perform both tasks if the locomotive is equipped with dynamic brakes. Electric motors are found in household appliances such as fans, exhaust fans, fridges, washing machines, pool pumps and fan-forced ovens.

Most electric motors work by electromagnetism, but motors based on other electromechanical phenomena, such as electrostatic forces and the piezoelectric effect, also exist. The fundamental principle upon which electromagnetic motors are based is that there is a mechanical force on any current-carrying wire contained within a magnetic field. The force is described by the Lorentz force law and is perpendicular to both the wire and the magnetic field. Most magnetic motors are rotary, but linear motors also exist. In a rotary motor, the rotating part (usually on the inside) is called the rotor, and the stationary part is called the stator. The rotor rotates because the wires and magnetic field are arranged so that a torque is developed about the rotor's axis. The motor contains electromagnets that are wound on a frame. Though this frame is often called the armature, that term is often erroneously applied. Correctly, the armature is that part of the motor across which the input voltage is supplied. Depending upon the design of the machine, either the rotor or the stator can serve as the armature.

DC motors for gate control

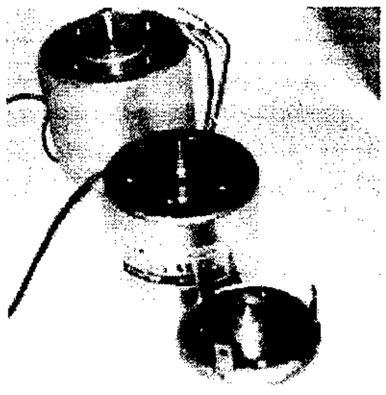
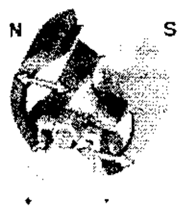


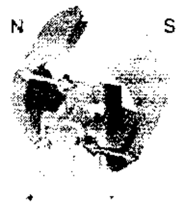
fig.11. DC motor

Electric motors of various sizes.

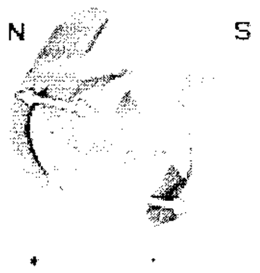
DC motor rotation



A simple DC electric motor. When the coil is powered, a magnetic field is generated around the armature. The left side of the armature is pushed away from the left magnet and drawn toward the right, causing rotation.



The armature continues to rotate.



When the armature becomes horizontally aligned, the commutator reverses the direction of current through the coil, reversing the magnetic field. The process then repeats.

If the shaft of a DC motor is turned by an external force, the motor will act like a generator and produce an Electromotive force (EMF). During normal operation, the spinning of the motor produces a voltage, known as the counter-EMF (CEMF) or back EMF, because it opposes the applied voltage on the motor. This is the same EMF that is produced when the motor is used as a generator (for example when an electrical load (resistance) is placed across the terminals of the motor and the motor shaft is driven with an external torque). Therefore, the voltage drop across a motor consists of the voltage drop, due to this CEMF, and the parasitic voltage drop resulting from the internal resistance of the armature's windings. The current through a motor is given by the following equation:

$I = (V_{applied} - V_{cemf}) / R_{armature}$

The mechanical power produced by the motor is given by:

$P = I * V_{cemf}$

Since the CEMF is proportional to motor speed, when an electric motor is first started or is completely stalled, there is zero CEMF. Therefore the current through the armature is much higher. This high current will produce a strong magnetic field which will start the motor spinning. As the motor spins, the CEMF increases until it is equal to the applied voltage, minus the parasitic voltage drop. At this point, there will be a smaller current flowing through the motor. Basically, the following three equations can be used to find the speed, current, and back EMF of a motor under a load:

$Load = V_{cemf} * I$

$V_{applied} = I * R_{armature} + V_{cemf}$

$V_{cemf} = speed * Flux_{armature}$

PIC PROGRAM

9. PIC program:

```
#include <lcd.h>
#include<pic.h>

#define LCD_LINE1 lcd_command(0x80);
#define LCD_LINE2 lcd_command(0xc0);
#define LCD_CLR   lcd_command(0x01);
#define SERIAL_NEXT_LINE serial_trans(0x0d);serial_trans(0x0a);

static volatile bit       RS  @ (unsigned)&PORTD*8+0;
static volatile bit       RW1  @ (unsigned)&PORTD*8+1;
static volatile bit       EN  @ (unsigned)&PORTD*8+2;

void serial_trans(unsigned char);
unsigned char serial_recev(void);
void serial_line_trans(unsigned char const*);

unsigned int take_adc(unsigned char);

void lcd_init(void);
void lcd_command(unsigned char);
void lcd_data(unsigned char);
void print_line(unsigned char const*,unsigned char);
void delay(unsigned long);
unsigned int st1,st2,st3,st4;

void main()
{
        TRISB=TRISD=0x00;
        TRISC=0xff;
        lcd_init();
        delay(100);
home:
        print_line("TRAIN COLLISION ",0x80);
        print_line("    SYSTEM    ",0xc0);
   RD4=0;
        RD5=0;
        RD6=0;
        RD7=0;
   delay(65000);
        print_line("                ",0xc0);
```

```
void lcd_init()
{
                  lcd_command(0x38);
                  delay(1000);
            lcd_command(0x38);
            delay(1000);
            lcd_command(0x38);
            delay(1000);
            lcd_command(0x38);
            delay(1000);

            lcd_command(0x0c);
            delay(1000);
                  lcd_command(0x06);
            delay(1000);

                  lcd_command(0x01);
            delay(1000);
            lcd_command(0x80);
            delay(1000);


}

void print_line(unsigned char const*lcd_disp.unsigned char loc_lcd)
{
         lcd_command(loc_lcd);
         while(*lcd_disp)
         {
         lcd_data(*lcd_disp);
         lcd_disp++;
         }
}


void lcd_command(unsigned char com)
{
            PORTB=com;
            RW1=0;
      RS=0;
                  EN=1;
            delay(100);
      EN=0;
      delay(400);
 }
```

```
void lcd_data(unsigned char dat)
{
                PORTB=dat;
                RW1=0;
                RS=1;
                EN=1;
                delay(100);
                EN=0;
                delay(300);
}

void delay(unsigned long del_del)
{
        while(del_del--);
}

do
{
        RD4=1;
        if(RC0==0)
        {
        do
        {
               if(RC0==0)
               {
               print_line("STATION A TO B    ",0x80);
               st1=1;
               RD4=1;
               RD5=1;
               RD6=0;
               RD7=0;                          //LED
               }

               if(RC1==0)
               {

               print_line("STATION B TO C    ",0x80);
               print_line("CLOSE THE GATE    ",0xC0);
               st2=1;
               RD4=1;                                   //CLOSE THE GATE

               RD5=1;
               RD6=1;
               RD7=0;
               }
if(RC2==0)
{
print_line("STATION C TO D    ",0x80);
print_line("OPEN  THE GATE    ",0xC0);
RD4=1;
RD5=0;
RD6=0;
RD7=1;
st3=1;

}

if(RC3==0)
{
print_line("STATION D TO E    ",0x80);
print_line("                  ",0xC0);
st4=1;
RD4=1;

}
}

while(RC3==1);
RD4=1;
while(RC3);
RD4=1;
print_line("STATION D TO C    ",0x80);
print_line("                  ",0xC0);

if(RC1==0)
{
RD4=1;
print_line("STATION B TO C    ",0x80);
st2=1;
}

while(RC2);
print_line("STATION C TO D    ",0x80);

RD4=0;
print_line("TRAIN STOPPED   ",0xc0);
delay(100000);
break;
}
```

```
if(RC3==0)
{
do
{
if(RC3==0)
{
print_line("STATION D TO C     ",0x80);
st1=1;
RD4=1;
RD5=1;
RD6=0;
RD7=0;                              //LED


}

if(RC2==0)
{
print_line("STATION C TO B     ",0x80);
print_line("CLOSE  THE GATE    ",0xC0);
st2=1;
RD4=1;
RD5=1;
RD6=1;
RD7=0;                         //CLOSE THE GATE

}

if(RC1==0)
{
        print_line("STATION B TO A     ",0x80);
        print_line("OPEN  THE GATE     ",0xC0);
        st3=1;
        RD4=1;
        RD5=0;
        RD6=0;
        RD7=1;                          //OPEN THE GATE
        }
if(RC0==0)
{
        print_line("STATION A TO Z     ",0x80);
        print_line("                   ",0xC0);
        st4=1;
        RD4=1;
```

```
         }

                  }


while(RC0==1);
print_line("STATION A TO B    ",0x80);
print_line("                  ",0xC0);
        RD4=1;

if(RC1==0)
{
print_line("STATION B TO A    ",0x80);
st2=1;
        RD4=1;
}

while(RC1);

RD4=0;
print_line("STATION C TO B    ",0x80);

print_line("TRAIN STOPPED   ",0xc0);
delay(100000);
break;
}
}while(1);
}
```

CONCLUSION

10. CONCLUSION:

The proposed design "Embedded system based Train collision avoidance system" is implemented with an RF Transceiver and PIC16F877A controller. This design aids to control and monitor the train position and gate control for unmanned railway crossing. This concept can also be implement using advance technology such as GPS, GSM with more accurate sensors.

This design avoids the collision of two trains in the same track and unmanned railway gate control. A 2-meter track is constructed and tested for the collision avoidance and gate control. This design will protect valuable human life.

FUTURE EXPANSION

11. FUTURE EXPANSION:

The proposed design is implemented by IR transceivers and RF module. This design will be implemented by high precision sensors. GSM and GPS technology. This design will be controlled by GSP technology with wireless technology, also monitored and control many tracks and trains. But GPS system is more expensive .

12. REFERENCE:

1.Mr.S.Raj Kumar . text book on Embedded system design. PHI publication. second edition.

2.www. Microchip.com

3. www. Sensors.com

4.www.google.com