Proceedings of the
International Conference on Mechanical Engineering 2005
(ICME2005) 28- 30 December 2005, Dhaka, Bangladesh

ICME05-AM-03

DESIGN AND FABRICATION OF FARE METER OF TAXICAB USING MICROCONTROLLER

Md. Tofiqul Islam

Department of Mechanical Engineering, Bangladesh University of Engineering and Technology, Dhaka 1000, Bangladesh. Email: mtislam@me.buet.ac.bd

ABSTRACT

Fare is one of the major expenses in public transport at everyday. So correct estimation of fare is important for passenger. In many vehicles, fare meters are used. Recently fare meters are built based on micro controller. Most of the builders use ATMEL 8051 micro controller, because this micro controllers are come into hand first. These micro controllers are CISC architecture. They have much more instructions But recently PIC well developed by MicroChip Company, is based on RISC architecture. In this paper, a fare meter has been built, based on PIC 16F84A, which is one of the most popular and versatile in use. Here the hardware and the software have been accomplished for a single and vital parameter i.e. bill of transport which is calculated by means of measuring the rotation of taxicab wheel and displayed by a display driver circuit. By measuring rotation, distance traveled by taxicab, time of travel and speed can also be calculated and displayed. This fare meter has been developed in respect of our country, Bangladesh to reduce cost of manufacturing and make it available in market. Thus this attempt will also reduce the import of these fare meters from other countries, especially from India.

Keywords: Fare meter, Taxicab, Microcontroller, Embedded system, PIC, Program loader

1. INTRODUCTION

The 1891 invention by German Wilhelm Bruhn of the taximeter (the familiar mechanical and now often electronic device that calculates the fare in most taxis) ushered in the modern taxi. (The "taxi" in "taximeter" is related to the word "tax," or "rate.") The first modern, gas-powered, meter-equipped taxi was the Daimler Victoria, built by Gottlieb Daimler in 1897; the first motorized taxi company began operating in Stuttgart the same year. For the distance travelled, fares for taxis are usually higher than for other forms of public transport (bus, tram, metro, train). The fare often does not depend on the number of people travelling together in a taxi. Sometimes there is a system where strangers share a taxi and fares are per person. Fares are usually calculated according to a combination of distance and waiting time, and are measured by a taximeter ("meter" for short and the origin of the word "taxi"). Instead of a metered fare, passengers sometimes pay a flat fare. In some countries, when demand is high--for instance, late at night--a taxi will pick up whoever offers the highest fare [1]. In early years, the fare of taxicab was estimated by measuring the distance of kilometer, which it was traveled. Say, TK. 3 per kilometer. That distance was measured by the analog device i.e. odometer. The odometer operated by a pair of gears from the speedometer shaft. Its motion was carried through the gears to the kilometer number rings. Then by multiplying the traveled distance and TK. per kilometer, total amount of fare was calculated. Now the measurement of fare has become easy with the rapid development of digital technology based on microcontroller. Bangladesh cannot give effort on the development of these fare meters; rather these meters are imported from other countries, though the technology of these fare meters is not tough one. Well knowledge about the microcontroller and the simulation of eletro-mechanical system can make it easy to build. Here a fare meter of taxicab has been developed which is an embedded system, controlled by microcontroller. This device has its own circuit design, which is depended on the programming logic. This device has been developed with a single parameter i.e. fare.

2. ABOUT MICROCONTROLLER

Microcontroller is the one of the ways of the evolution of microprocessor. It consists of a microprocessor, RAM, EEPROM or EPROM, I/O capacities, ADC, timer, interrupt controller and embedded controller. The microcontroller chip has the versatility to sense inputs and control outputs in the devices. The rapid development of microcontroller is happening due to its low cost, versatility, ease of programming and small size. The most popular microcontrollers are Intel 4048, Motorola AT89C51, AT80C81, Microchip's PIC. Here Microchip has been preferred because it has RISC architecture, wide acceptance in industry, huge

information resources, low cost and ease of use. RISC microcontroller has the HARVARD architecture rather than VON-NEUMAN due to it fetch instruction in a single cycle (all 14 bits) a separate bus allows one instruction to execute while the next instruction is fetched.

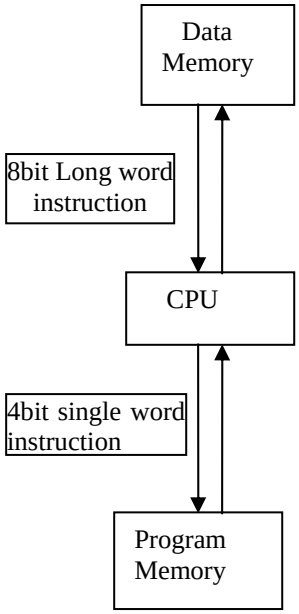

Fig 1. Harvard Architecture

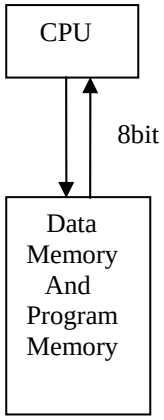

Fig 2. Von-Neuman Architecture

2.1 Selection of Pic16f84a

F-family has been selected because it can be burned 1 million times without enabling code protection option. PIC 16F84A has the following features [2]:

1. Mid range and 16 series
2. Low cost 8 bit microcontroller
3. Flash type (have EEPROM flash memory for program and data storage)
4. No built in ADC, DAC, Serial communication capacity.
5. Has 13 I/O (Bi-directional lines)
6. Less instruction
7. Instructions are in a well-defined form

2.1.2 CPU RISC Features [2]

1. 1024 words of program memory
2. 68 bytes of Data RAM
3. 64 bytes of Data EEPROM
4. 14 bit wide instruction words
5. 8 bit wide data bytes
6. 8 bit=1 byte
7. Stack level =8
8. Four interrupt sources:
   (i) External interrupt RB0
   (ii) TMR0 timer overflow
   (iii) RB4~RB7 pins for internal interrupt
   (iv) Data EEPROM write complete

2.1.3 Peripheral Features [2]

1. 13 I/O pins (bi-directional)
2. 25mA (max) per pin
3. TMR0: 8 bit timer/counter with 8 bit programmable prescaler

2.1.4 Special Features [2]

1. Power-on Reset (POR), Power-up Timer (PWRT), Oscillator Start-up Timer (OST)
2. Watchdog Timer (WDT)
3. Code Protection
4. Selectable oscillator options (XT, RC, HS)

2.2 Memory Mapping

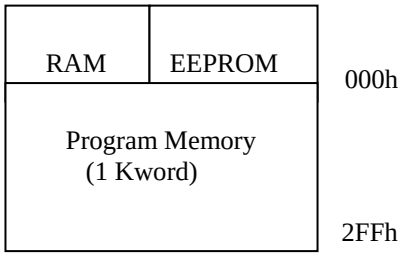

Fig 3. Memory mapping.

Table 1: Register File Map

PIC16F83/16CR83	PIC16F84/16CR84
General Purpose register (SRAM), 36 bit, 0h~2Fh	General purpose register (SRAM), 64 bit, 0Ch~4Fh
Mapped (accesses) in Bank0, 8Ch~4Fh	Mapped (accesses) in Bank0, 8Ch~CFh

2.3 Programming

The following steps are to maintain for the PIC programming. The steps are:

Step1: Declaration

Addressing special function registers and variable. For example, TMR0 equ 02h, num equ 0ch

Step2: Definition

Defining anyone bit of 8 bit register. For example, #define pa status,5

Step3: Initialization

Initialize whether interrupt may be present or not, port configuration setting according to byte or bit wise. For example, clrf trisa

Step4: Program

Main program code

3. PC DRIVEN PROGRAMMER

Microcontroller chipmaker, Microchip technology Inc. has been phenomenally successful with its low-cost PIC micro family in the last few years. PICs are used in embedded in high proportion of smart card. The popular programmer is run by PC. Microcontroller can also be burned by pressing keys on a keypad. Here a programmer has been developed that is run on PC with the help of EPICWIN software. This PC driven programmer is the easiest way of writing and erasing the PIC.

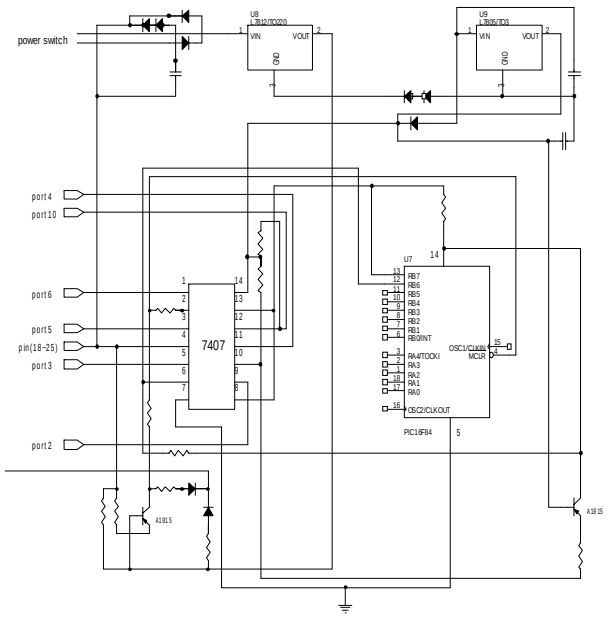

Fig 4. Circuit diagram of program loader

3.1 Working Principle of the PIC Programmer

The working principle of the PIC programmer is very simple. First, the programmer has been power by 15V AC transformer. A bridge rectifier has been used to protect against the reverse polarity damage and also to the rectify incoming AC [3]. A regulator 78L05 has been used to provide 5V Vdd supply rail while 78L12 has been used to provide 12V Vpp rail. (Another method is that a regulator 78L05 has been used to provide 5V Vdd while 78L07 has been piggybacked on this 5V rail to provide 12V Vpp rail.)

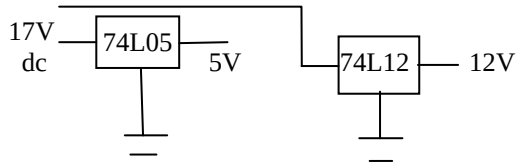

Fig 5. Method 1

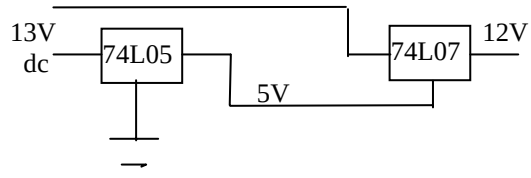

Fig 6. Method 2

Interfacing between the microcontroller chip socket and the PC's printer port has been done via 74LS07 hex inverter. This allows the PC software to control the Vdd switching via pin 5 of the printer port, inverter and the transistor A1015. Similarly the Vpp switching has been controlled via pin 4 of the printer port, inverter and another transistor A1015 [3]. In addition, programmer clock pulses have been sent to the chip socket via pin 3 of printer port, inverter, while the programming data has been sent to the chip via pin 2 of printer port and inverter. Finally it has been read data from the chip's EEPROM via inverter and acknowledgement pin (pin 10) of the printer port.

4. FARE METER OF TAXICAB

Fare meter has been developed because it is not manufactured in Bangladesh. In general, it is imported from India which is expensive (roughly TK 6000) and replaced after 6 months. Here PIC16F84A has been used to develop the fare meter which has RISC architecture (only 35 instructions).

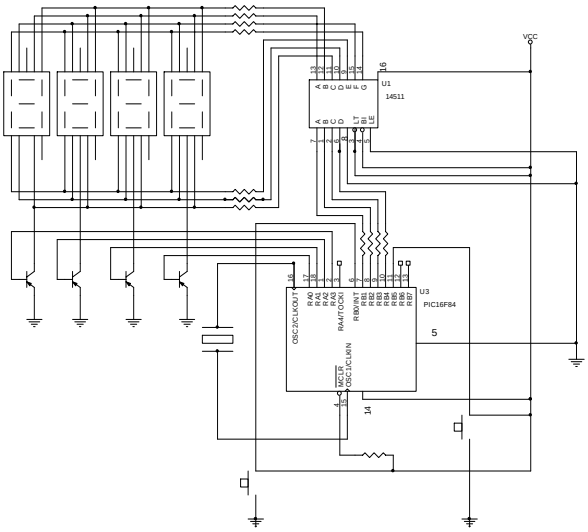

Fig 7. Circuit diagram of fare meter

4.1 Working Principle of Fare Meter of Taxicab

In fig 7 push button is assumed to be bounce free, implying that when it is pressed and then released, a single pulse is produced (the signal goes high when it is pressed and goes low when it is released.) When push button at RB5 is pressed, RB5 becomes high and microcontroller receives this signal and processes it according to the user program. After pressing the push button three times, microcontroller will send 0 bit at RA0 to the base of the PNP transistor while other pins (RA1, RA2, RA3) remaining the same. Then PNP transistor will switch the common pin of the 1st seven-segment to the ground. Thus this seven segment will be active and simultaneously microcontroller will send 4 bit data to the decoder through RB1~RB4. Decoder IC will decode the data into 7 bit data and display the seven-segment. Thus after 3 subsequent input signals microcontroller will increase the bill by TK 1.00.

Again when there is no input signal created by push button, the microcontroller will be in waiting mode. Thus after waiting for 1.25 minute, microcontroller will increase the bill by TK 1. This process will be continued until the fare meter is reset.

The scanning of the four digits over four seven segments has been so fast that it can not be sensed by eye

vision. Here the scanning time between the subsequent two digits, which has been done by the microcontroller, is 2.311 milliseconds under the operation of 4 MHz XT crystal. The sensing time from eye to brain is minimum 0.1 second, which is greater than the scanning time. So it seems that the four digits are lighted steadily. Assembly code of this fare meter is mentioned in table 4.

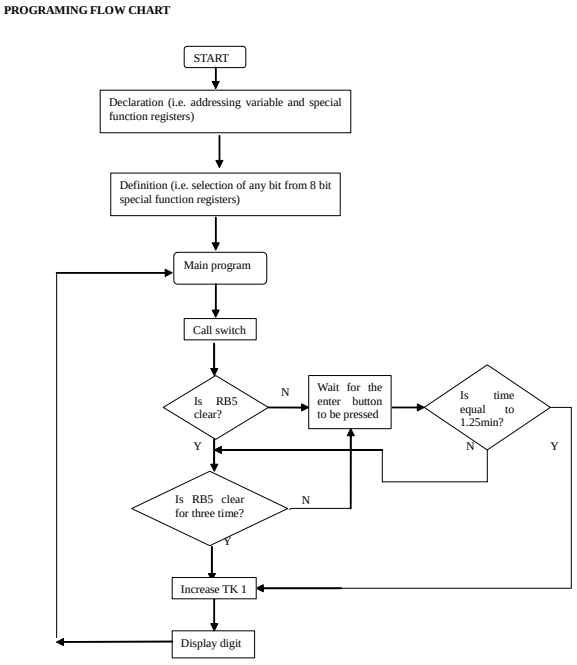

Fig 8. Flow chart of programming

Table 2: Cost Estimation of Fare Meter

Name of components	Quantity	Cost
PIC16F84A	1	220 TK.
Decoder (MC14511B)	1	110 TK.
Crystal (4MHZ)	1	30 TK.
PNP transistor (A1015)	4	15 TK. (each)
Push button	2	5 TK. (each)
10K	2	1 TK. (each)
1K	14	1 TK. (each)
Socket	2	20 TK. (each)
7 segment display	4	10 TK. (each)
Vero board	2	25 TK (each)
Total		576 TK

Table 3: Cost Estimation of Programmer

Components	Quantity	Cost
Semiconductor		
PIC16F84A	1	220 TK
74LS07	1	70 TK
78L05	1	10 TK
78L12	1	12 TK
A1015	2	15 TK (each)
LED	2	1 TK (each)
Diode	7	1 TK (each)
Resistors		
1K	4	1 TK (each)
2K	4	1 TK (each)
10K	2	1 TK (each)
Capacitors		
100mF 25V	2	10 TK (each)
1000mF 50V	1	15 TK
Miscellaneous		
PCB	1	60 TK
IC socket (18 pin)	1	20 TK
IC socket (16 pin)	1	20 TK
Parallel port	1	60 TK
Power switch	1	5 TK
Power cord	1	10 TK
Total		600 TK

5. RESULTS AND DISCUSSION

This fare meter of taxi cab has been working properly but not precisely. Fare has been displayed both for the distance covered during running time and the waiting time. Debouncer has not been used here. But if this same device is connected with the opt coupler in stead of push button and a rotating disk having a aperture near the periphery, is run through the opt coupler , then the device will work properly without using of debouncer. Bangladesh imports these fare meters from India which cost 6000 TK per meter and the durability of these meters are low. But this fare meter of taxicab, which has been mentioned in this paper, costs only TK 576. If this meter is made commercially then it may cost TK 2000. Also the program loader, which is cost TK 7000, can be built commercial within TK 1300. When this meter will be built commercially then “interrupt” instruction will be used in the program logic.

6. CONCLUSIONS

The application of microcontroller in fare meter has been focused in this paper. The complete product of faremeter needs some modification (i.e. redesigning the deriver circuit for displaying distance and waiting time). The Circuit design of this fare meter reduces the number of four decoder ICs (MC14511B) to one for displaying four digits. Thus the common of each of the four seven segment has been grounded by turn. If four decoder ICs are used, less I/O pins are required to use and latch enable of the decoders has to do. The use of microcontroller has been made the driver circuit easy since the control part of the fare meter has been accomplished by program. In Bangladesh, commercial support can make it to be a real life application in public transport.

7. REFERENCES

1. http://en.wilipedia.org/wiki/Taxicab
2. MICROCHIP PIC16F84A DATASHEET ©2001 Microchip Technology Inc.
3. Kit81 Simple PIC MICRO Programmer / Experimenter http://www.kitsrus.com
4. Peatman, J.B., 2002, D*ESIGN with PIC MICROCONTROLLER*, Pearson Education Asia, ISBN81-7808-508-9.

Table 4: Assembly code of fare meter of taxicab

1 porta equ 05	39 delay	77 movwf porta	115 movlw .10
2 portb equ 06	40 movlw .1	78 call delay	116 movwf num1
3 trisa equ 85h	41 movwf count	79 movf sseg2,0	117 L3
4 trisb equ 86h	42 time1	80 call table	118 movlw .10
5 status equ 03	43 clrf tmr0	81 movwf portb	119 movwf temp
6 tmr0 equ 01	44 btfss tmr0,7	82 movlw .251	120 call switch
7 pcl equ 02	45 goto $-1	83 movwf porta	121 gate
8 count equ 0ch	46 decfsz count,f	84 call delay	122 movf temp,0
9 wait equ 31h	47 goto time1	85 movf seg3,0	123 decfsz temp,1
10 wait1 equ 32h	48 retlw 00	86 call table	124 goto $-3
11 wait2 equ 33h	49 init	87 movwf portb	125 vital
12 wait3 equ 34h	50 bsf pa0	88 movlw .253	126 call display
13 seg1 equ 0dh	51 movlw .16	89 movwf porta	127 incf seg1,1
14 seg2 equ 0eh	52 movwf trisa	90 call delay	128 decfsz num1,1
15 seg3 equ 21h	53 movlw .33	91 movf seg4,0	129 goto L3
16 seg4 equ 25h	54 movwf trisb	92 call table	130 movlw 00
17 num1 equ 17h	55 movlw .131	93 movwf portb	131 movwf seg1
18 num2 equ 18h	56 movwf optreg	94 movlw .247	132 incf seg2,1
19 num3 equ 20h	57 bcf pa0	95 movwf porta	133 decfsz num2,1
20 num4 equ 26h	58 goto main	96 call delay	134 goto L4
21 stay equ 21h	59 switch	97 retlw 00	135 movlw 00
22 temp equ 22h	60 movlw .10	98 main	136 movwf seg1
23 #derfine pa0 status,5	61 movwf wait1	99 movlw .255	137 movwf seg2
24 optreg equ 81h	62 loop1	100 movwf porta	138 incf seg3,1
25 org 00	63 movlw .256	101 movwf portb	139 decfsz num3,1
26 goto init	64 movwf wait	102 movlw 00	140 goto L5
27 table	65 loop	103 movwf seg1	141 movlw 00
28 addwf pcl,f	66 call display	104 movwf seg2	142 movwf seg1
29 retlw .00	67 btfss portb,5	105 movwf seg3	143 movwf seg2
30 retlw .1	68 goto gate	106 movwf srg4	144 movwf seg3
31 retlw .2	69 decfsz wait1,1	107 movlw .10	145 incf seg4,1
32 retlw .3	70 goto loop1	108 movwf num4	146 decfsz num4,1
33 retlw .4	71 goto vital	109 L6 movlw .10	147 goto L6
34 retlw .5	72 display	110 movwf num3	148 nop
35 retlw .6	73 movf seg1,0	111 L5	150 goto main
36 retlw .7	74 call table	112 movlw .10	151 end
37 retlw .8	75 movwf portb	113 movwf num2	
38 retlw .9	76 movlw .254	114 L4	