

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320-088X

IJCSMC, Vol. 3, Issue. 10, October 2014, pg.528 – 535

RESEARCH ARTICLE



Automation of Testing Using a Microcontroller

Harman Malhotra

Department of Electronics and Communication

Bhagwan Parshuram Institute of Technology

New Delhi, India

Harmanmalhotra2000@gmail.com

Abstract— *Testing is very important for any company before launching its product in the market. Manual testing becomes both taxing and monotonous and after a point errors begin to creep into the products being tested. In the paper presented a method to automate the overall testing process is discussed. A remote controlled car is interfaced with a microcontroller and is tested using a computer; hence, the manual angle is eliminated out of the overall testing process.*

The paper highlights just one of the many possible ways in which testing can be automated, and the concept discussed can be expanded to meet industrial demands.

Keywords — *Microcontroller, Voltage Regulator, Line Driver, Darlington Pair*

I. INTRODUCTION

The automated testing system consists of 2 sections, which are as follows:

The *receiving section* consists of a remote. There are various standards of the car remote systems used in the market. In this project a remote of 40MHz frequency is used. Therefore, whenever a key is pressed on the remote a coded pulse train is emitted. The pattern of the pulse train depends on the key pressed.

The *transmitting section* consists of the 89c2051 microcontroller. The microcontroller is programmed so as to decode the incoming coded pulse train which is sensed by a sensor. Once the coded pulse train is received it is decoded to recognize the corresponding car for which the key is pressed and the necessary action is taken. We can have four command buttons (i) 'Forward' for movement of car in forward direction (ii) 'Backward' for movement of car in backward direction (iii) 'Right' for turning car in right direction and (iv) 'Left' for turning car in left direction.

II. SYSTEM MODEL

As described in the introduction section, the system is divided into the transmitter and receiver section. The Computer communicates with the object being tested via the microcontroller. The Vcc for the microcontroller is provided by a voltage regulator circuit built using 7805, this also provide the required source voltage for MAX232. The MAX232 acts as the line driver required to convert TTL compatible signals into RS232 compatible signals that the computer can understand.

Each relay coil is connected to one of the four direction buttons on the remote, the microcontroller is however, unable to provide sufficient amount of current to energize the relay coils. Hence, ULN 2003 is used to drive the relay coils.

Hence, when a command button is pressed on the computer, the object being tested (in this case a remote controlled car) performs the required action. This results in the automation of the overall testing process.

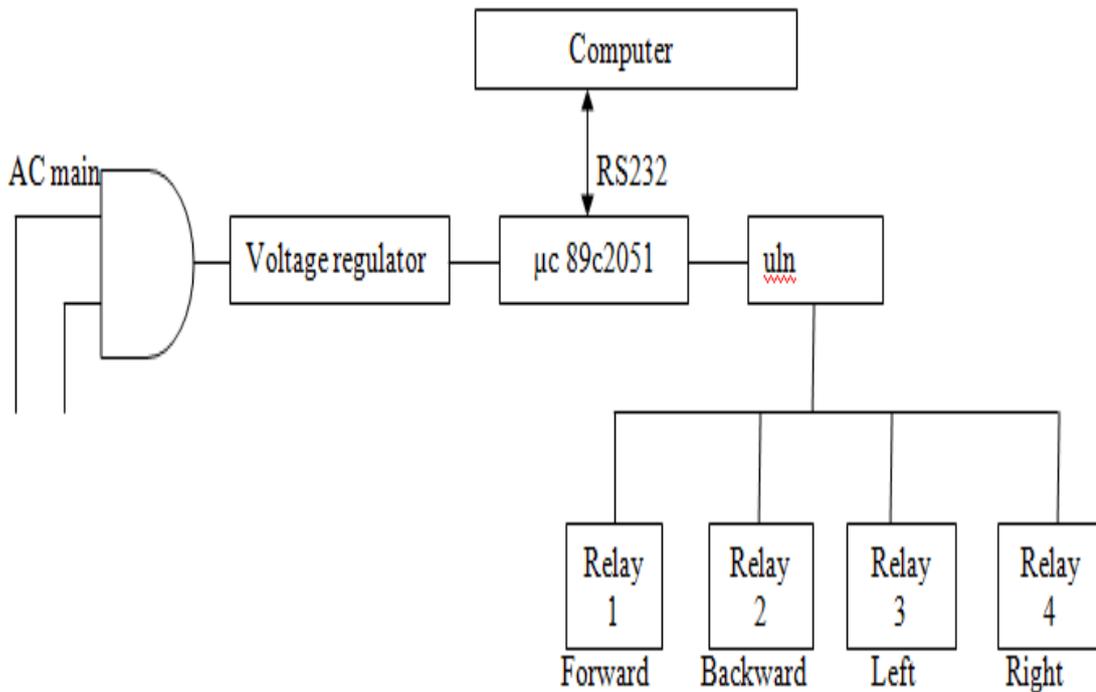


Fig 1: System Model

III. VOLTAGE REGULATOR CIRCUIT

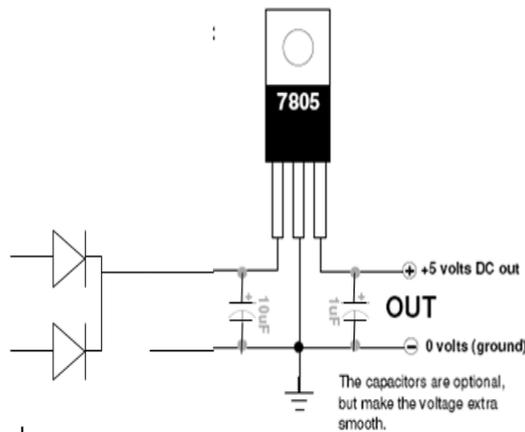


Fig 2: Voltage Regulator Circuit [1]

The 7805 is a VOLTAGE REGULATOR. It looks like a transistor but it is actually an integrated circuit with 3 legs. These legs are numbered 1, 2 and 3 starting from the left. One stands for input, 2 for ground and 3 for output. The capacitors of 1uF and 10uF are suggested by the the output and the internal reference voltage; it however becomes unstable if a variation in supply voltage causes changes in the output voltage.

IV. AT89C2051 MICROCONTROLLER and it's CONNECTIONS

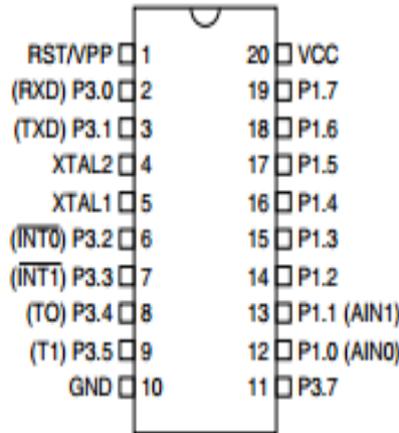


Fig 3: AT89C2051 PIN Diagram [2]

The AT89C2051 is a low-voltage, high-performance CMOS 8-bit microcomputer with 2K bytes of Flash programmable and erasable read-only memory (PEROM). The AT89C2051 provides the following standard features: 2K bytes of Flash, 128 bytes of RAM, 15 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, a precision analog comparator, on-chip oscillator and clock circuitry [2]. The crystal oscillator of frequency 11.0592 MHz is connected across pin no 4 & 5 of 89c2051 microcontroller. A capacitor of 220 µf connected between pin no 1 of microcontroller and 5v supply. 1kΩ resistor connected between pin no 1 and 20 of microcontroller. A 5v supply connected at pin no 20 and ground connected at pin no 10 of microcontroller. pin no 16,17,18 and 19 of microcontroller are connected with pin no 4,3,2 and 1 of uln2003 respectively. These four connections shows the four directions of the car forward, backward, left and right.

V. ULN 2003 and it's CONNECTIONS

The ULN is a type of Darlington transistor pair which has very high current gain and acts a driver to energize the coils of the relay. The ULN is used because the pins of the microcontroller cannot provide sufficient current to drive the relay. 12v vcc is given at pin no 9 of uln and pin no 8 is grounded. Pin no 13 of uln is connected to fist relay which controls forward direction of the car. Similarly 14, 15and 16 are connected to relay no 2,3 and 4 which controls backward, left and right directions of the car respectively.

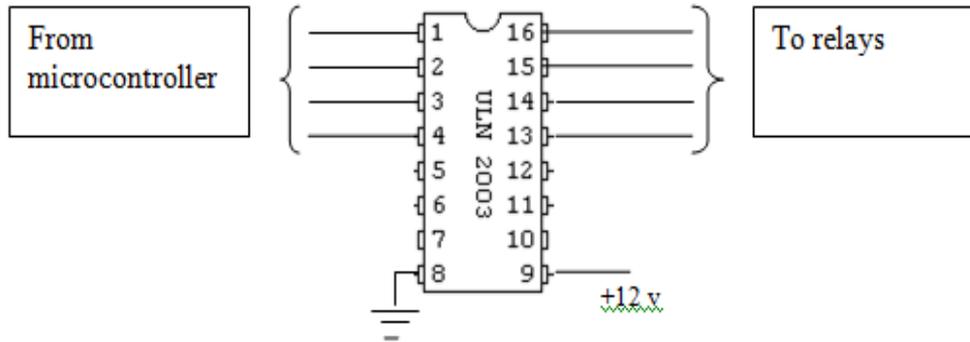


Fig 4: ULN2003 Connections [3]

VI. RELAY



Fig 5: Relay Coil Used [4]

A relay is nothing but an electronically controlled switch, it allows the isolation of two sections of a system using two separate voltage sources.

Relays allows a device to effortlessly control a much larger device, allow a very small current to control a large current device.

VII. MAX232 and it's CONNECTIONS

The TxD and RxD pins used for transmission and reception of data serially by the microcontroller are TTL compatible. Hence, the need of a line driver (voltage converter) arises to make them RS232 compatible.

MAX232 converts from TTL voltage levels to RS232 voltage levels. The advantage of using MAX232 is that it requires a +5V source which is same as that for the microcontroller.

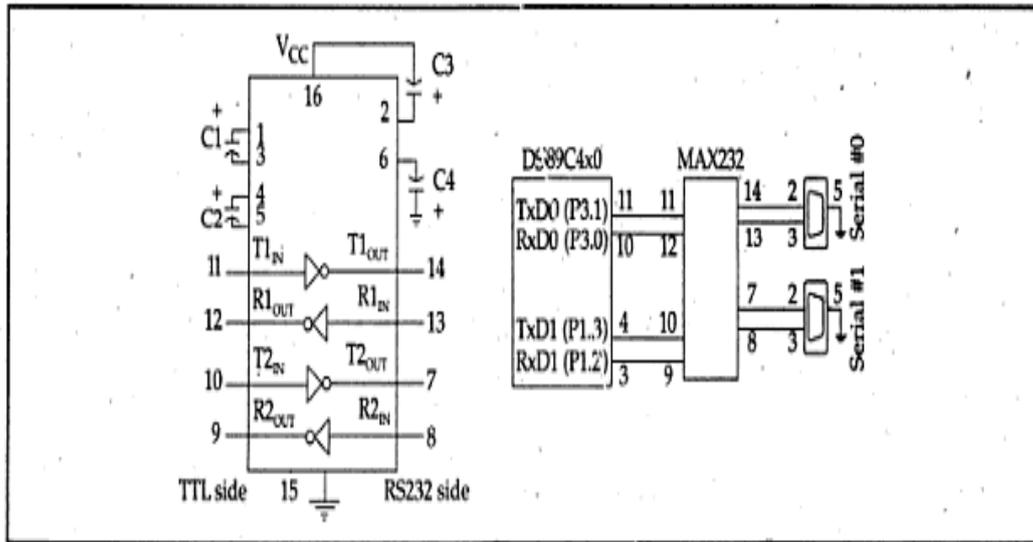


Fig 6: MAX232 Connections [5]

VIII. ASSEMBLY LANGUAGE PROGRAM

```

;*****
;
; CPU 8051
bdr: equ 256-6 ;Baud rate=9600
psw: equ 0D0H
a: equ 0E0H
sp: equ 081H
pcon: equ 087H
tcon: equ 088H
tmod: equ 089H
P0: equ 080H
P1: equ 090H
P2: equ 0A0H
P3: equ 0B0H
TL1: equ 08BH
TH1: equ 08DH
scon: equ 098H
sbuf: equ 099H
dpl: equ 082H

dph: equ 083H
TLO: equ 08AH
TH0: equ 08CH
IE: equ 0A8H
IP: equ 0B8H
B: equ 0F0H
scon0: equ 098H
scon1: equ 099H
tcon6: equ 08EH
;*****ASCIIcodes*****
; A:041H B:042H C:043H D:044H E:045H F:046H G:047H H:048H I:049H
; J:04AH K:04BH L:04CH M:04DH N:04EH O:04FH P:050H Q:051H R:052H
; S:053H T:054H U:055H V:056H W:057H X:058H Y:059H Z:05AH
; *:02AH +:02BH ,:02CH -:02DH .:02EH /:02FH
    
```

```

;::03AH ;:03BH <:03CH =:03DH >:03EH ?:03FH @:040H
;*****
org 0000H
strt: mov sp,#065H ;set stack ptr=65
      mov pcon,#080H ;select SMOD=1
      mov tmod,#022H ;select both contrs as timers RCLK=0
      mov TH1,#0FAH ;select baud rate to 9600
      mov TL1,#0FAH ;set TL1=TH1
      setb tcon.6 ;start counter 1
      mov scon,#52H ;MODE=1, REN=1, TI=1, RI=0
      mov a,#0FFH ;set 0FF to a
      setb tr1
      mov a,#00h
      mov P1,#00h
      mov r7,#00h

12: acall recv
      mov r7,a
      cjne a,#'F',11
      mov P1,#080h
      ajmp 12

11: nop
      mov a,r7
      cjne a,#'B',15
      mov P1,#040h
      ajmp 12

15: nop
      mov a,r7
      cjne a,#'L',13
      clr P1.4
      setb P1.5
      ajmp 12

13: nop
      mov a,r7
      cjne a,#'R',16
      clr P1.5
      setb P1.4
      ajmp 12

16: nop
      mov a,r7
      cjne a,#'S',17
      mov P1,#00h
      ajmp 12

17: nop
      mov a,r7
      cjne a,#'Z',18
      acall test

18: nop
      ajmp 12
;*****
recv: jnb ri,recv

```

```

mov  a,sbuf
clr  ri
ret
;*****
;return test data on <Z>
test: nop          ;no operation
      mov  a,#054H  ;mov 'T' to a
      acall snd     ;call send data
      mov  a,#045H  ;mov 'E' to a
      acall snd     ;call send data
      mov  a,#053H  ;mov 'S' to a
      acall snd     ;call send data
      mov  a,#054H  ;mov 'T' to a
      acall snd     ;call send data
      ret          ;return
;*****data sending routine*****
snd:  nop
      jnb  scon1,snd ;wait until last chr gone
      clr  scon1    ;TI=0
      mov  sbuf,a   ;Start transmit
      ret          ;return
;*****

```

Visual basic program:-

```

Dim Tr1 As String, X3 As Single
Dim Dstat1 As String

```

```

Private Sub back_Click()
Tr1 = "B"
Dstat1 = "BACKWARD"
lbl1.Caption = Dstat1 & Tr1
DoEvents
Call StSnd
End Sub

```

```

Private Sub Form_Load()
MSComm1.CommPort = 1
MSComm1.Settings = "9600,N,8,1"
If MSComm1.PortOpen = True Then MSComm1.PortOpen = False
MSComm1.PortOpen = True
End Sub
Private Sub Command7_Click()
End
End Sub

```

```

Private Sub fwd_Click()
Tr1 = "F"
Dstat1 = "FORWARD"
lbl1.Caption = Dstat1 & Tr1
DoEvents
Call StSnd
End Sub
Private Sub lft_Click()
Tr1 = "L"
Dstat1 = "LEFT"

```

```
lbl1.Caption = Dstat1 & Tr1
DoEvents
Call StSnd
End Sub
Private Sub DtSend()
MSComm1.CommPort = 1
MSComm1.Settings = "9600,N,8,1"
MSComm1.PortOpen = True
If MSComm1.PortOpen = True Then MSComm1.PortOpen = False
End Sub
Private Sub StSnd()
If MSComm1.PortOpen = True Then MSComm1.PortOpen = False
MSComm1.PortOpen = True
DoEvents
MSComm1.Output = Tr1
lbl1.Caption = Dstat1
DoEvents
If MSComm1.PortOpen = True Then MSComm1.PortOpen = False
End Sub
Private Sub rgt_Click()
Tr1 = "R"
Dstat1 = "RIGHT"
DoEvents
Call StSnd
End Sub

Private Sub stp_Click()
Tr1 = "S"
Dstat1 = "CAR STOPPED"
DoEvents
Call StSnd
End Sub
```

REFERENCES

- [1] <https://www.fairchildsemi.com/datasheets/LM/LM7805.pdf>
- [2] <http://www.atmel.com/Images/doc0368.pdf>
- [3] <http://www.ti.com/lit/ds/symlink/uln2003a.pdf>
- [4] <http://pdf1.alldatasheet.com/datasheetpdf/view/308942/OENINDIA/57DP-12-1C6.html>.
- [5] The 8051 Microcontroller and Embedded Systems: Using Assembly and C By Mazidi Muhammad Ali.