Driverless metro train with obstacle detector engineering essay

INTRODUCTION

This project is designed to understand the technology used in driver less metro train. This technology is the future of the metro train. This train will be equipped with the microcontroller as the CPU, Sensors, Displays, and Motors. These trains will be designed for following the specific path. This microcontroller will be programmed such that door opens and close automatically when the train reaches its station. This microcontroller will be programmed for a specific path so that it knows the distance between two stations. It also opens the door when train reaches the station for a certain time and then closes. Also if the train is following its path and in front of train an obstacle is detected the sensor will sense the obstacle from a safer distance so that the train can be stopped before accident.

In this project we are using the same pattern described above. Here we are using AT89S52 microcontroller as a CPU. One stepper motor is used for the train so that it can travel between the stations. Second, motor is used for the door open and close. For displaying the number of station and emergency message we are using 2 line LCD displays. Here the train is designed for ten stations numbered 1 to 10. LCD is used to display the message as station number reached. Also buzzer is used as a sound indicator of emergency alarm and door open. Sensor is used to detect the obstacle in front of the train. When the obstacle is detected the buzzer will sound , all the motors stop, and message will be displayed on the LCD. When the obstacle is removed the train will again start moving on its path.

The train equipped with microcontroller controls

LCD

L293D motor driving IC

PIR sensor

Buzzer

Led

LIST OF COMPONENTS

1 IC 8051 MC (AT89S52) – 1

2 IC L293D – 1

3 Transformer (120 V to 12V) – 1

4 Voltage Regulator 7806 – 1

5 Voltage Regulator 7812 – 1

6 2 X16 LCD display – 1

7 Stepper Motor – 2

8 Crystal Oscillator (11. 0592 MHz) – 2

9 Switch – 1

10 LED – 1

11 Resistors (1k, 10k, 8. 2k, 1×8 SIP10kÎ©)- pack of each

12 Capacitors(22pf, 10uf, 1mf, 10nf) – pack of each

13 Diode IN4007 – 4

14 Buzzer – 1

15 Transistor (5474A) – 1

16 PIR – 1

WORKING OF PROJECT:

Driver less metro train project is typically based on microcontroller. Where microcontroller used as a CPU is controlling the entire circuit. The micro IC is programmed to follow the sequence of instruction or interrupts generated during the whole operation. The whole project circuit is divided into different parts so that it is easy to understand and debug. The main parts are

Main microcontroller circuit

L293D motor driving circuit

Power supply circuit

LCD circuit

PIR sensor circuit

These four parts are connected together to make the whole project functional.

BLOCK DIAGRAM

led

POWER SUPPLY

MICRO

8051

L293D

IC

MOTOR

SENSOR

PIR

LCD 2X16

BUZZER

CIRCUIT DESIGN

(MULTISIM DESIGN, ULTIBOARD DESIGN)

POWER SUPPLY

For explaining the working of the project firstly we explain the architecture of the power supply. For this project we need the power supply that can convert 120V to 12V and 6 V. the main components used in the power supply are:

DC POWER SUPPLLY

Transformer (120V to 12V Step down, 0. 5Amp)

IN4007 diodes connected I bridge shape

Voltage regulator 7806 and 7812

Capacitor (1uf, 10 nf)

Resistor (10 kohm)

The 120 volts is step down by 12 volts by transformer. Then this 12V is rectified by the bridge rectifier made up of diodes. This bridge rectifier is used to provide DC voltage. Then the 12V is regulated by 7806 to 6V and 7812 to 12V. 1 micro farad capacitor is used to filter the pulsating DC voltage in order to get the pure DC. Here 6V is given to the microcontroller circuit. Other, 12V is given to the motor driving circuit. Here Voltage regulator is used so that no fluctuation occurs in order to get constant voltage. The LED attaches to check the correctness of power supply.

PIR SENSOR

PIR stands for Passive infrared sensor which detects the motion of the obstacle in front of the train. Its pin 1 is connected to the VCC, pin 3 connected to the ground and pin 2 is connected to the base 0 of the transistor. The output of the sensor is active high which is given to the base of the transistor. This transistor act like a not gate and convert active high input to active low output of the transistor output. The collector of the transistor is connected to the microcontroller port 3. 0. Resistance is connected to the base of the transistor to prevent overflow of the current. The output of the PIR sensor is digital. When PIR sensor senses the obstacle in front of the it sends active low input through transistor to the microcontroller as a interrupt. When microcontroller gets this interrupt the programming in the microcontroller stops all the motor bringing the train to stop. Also the led will glow and buzzer will sound showing the presence of obstacle.

LCD MODULE

In our project we are using 2 line and 16 digit LCD display. This lcd has 16 pins. It is connected as a parallel port to the microcontroller. Here we are using 14 pins two pins are not being used. Out of these 14 pins, 8 pins are used for data lines to communicate with the microcontroller. These 8 pins are connected to the port 2. 0 to port 2. 7 of the microcontroller. When the message is to be displayed by the microcontroller it is transferred through these data pins. Here data bit D7 is used for showing the busy message. When D7= 1 the LCD is busy and when it is D7= 0 than it is not. RW, EN, and RD are connected to the port 3 pins P3. 5 to P3. 7. EN pin is used to detect the high to low pulse. RS= 1 is used for register select that is data register to display data coming from microcontroller and RS= 0 command register to get command from microcontroller like clrscr. For data register RS = 1 and for command register RS= 0. For RW we have R= 1 and W= 0. VCC pin is connected to 5V power supply, GND pin is connected to ground. CV pin is connected to potentiometer which is used to adjust the contrast of the LCD.

MOTOR DRIVING CIRCUIT

The motor driving circuit depends upon the output of the microcontroller. 2 motors are connected to L293D IC. This IC is used separately to drive motor because the microcontroller voltage is low to drive the motors. L293D act as a amplifier for driving the motor. Here the output of the IC is same as the input the difference is just that the input voltage is amplified. Two motors are connected to the output of the IC. This L293D IC can rotate motor in either direction. L293D is connected to the port0 of the microcontroller through pull up resistor. In the staring both the motor are in off state. Each motor has two pins that are connected to the IC. Here instead of using both motor we are using CD tray motor that will act like a gate which will open when it get open command from microcontroller and close with the close command. we are using stepper motor that is a electro-mechanical rotary actuator that converts electrical pulses into unique shaft rotations. This rotation is directly related to the number of pulses. This motor can be operated in forward/reverse with controllable speed from a microcontroller through a transistor driver circuit. This stepper motor is a (4 phase) 12 VDC, 150 mA motor that takes 3. 6 degrees per step. Phase resistance (Ohms): 75 · Current (mA): 150 · Phase Inductance (mH): 39 · Detent torque (g-cm): 80 · Holding Torque (g-cm): 600 · Mounting hole space diagonal (in.): 1. 73 · Mounting hole (in.) 0. 11 · Shaft diameter (in.): 0. 197 · Shaft length (in.): 0. 43 · Motor Diameter (in.): 1. 66 ·Motor height (in.): 1. 35 · Weight: 0. 55 lbs.

BUZZER

C: UsersAdminDesktopimages (1). jpg

In this project buzzer is used as a alarm indicator. Buzzer has 2 pins its positive is connected to the supply voltage and negative pin is connected to microcontroller at port 1 pin p1. 0. Buzzer will sound when it receive 0 from the microcontroller and stop when it receive 1 from P1. 0. This buzzer will sound when the train reaches the station and also in case of emergency when obstacle is detected.

AT89S52

C: UsersAdminDesktopAT89S52-MICROCONTROLLER-configuration. jpg

The AT89S52 is a low-power, high-performance CMOS 8-bit microcontroller with 8K bytes of in-system programmable Flash memory. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with in-system programmable Flash on a monolithic chip, the Atmel AT89S52 is a powerful microcontroller which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89S52 provides the following standard features: 8K bytes of Flash, 256 bytes of RAM, 32 I/O lines, Watchdog timer, two data pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S52 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. In this microcontroller 11. 0592 crystal oscillator is used to produce clock pulses for the microcontroller functioning. These clock pulses are used for R/W cycle. 2 ceramic capacitor of 22pf are used to provide pure pulse to the microcontroller. A combination of resistance 10 kilo ohm and capacitor of 10uf are used to provide reset circuitry. When we provide power to microcontroller it should be reset for some time. The reset time is given by t= 1. 1*R*C. This reset circuitry is connected to port 1 reset pin; this circuit can be manually reset using switch which is also connected to the reset pin. Interrupt pin INTO is connected to the output of the transistor which gives active low input as a interrupt. Port 0 is connected to 1×8 (10 kohm) SIP resistors to provide the external pull up. Port is used for the output to the motor driver circuit. Port 2 is connected to the LCD module.

TROUBLESHOOT

1-Care should be taken while choosing the components with proper rating.

2- Power supply should maintain preventing short circuit.

3- Project should be handled with care since IC is delicate.

4- Project circuit should be simple so that it is to debug.

5-Etching and drilling holes should be done carefully.

6-Soldering should be done with proper precaution

Advantages

1-Lower expenditure for staff . However, service and security personnel is common in automated systems.

2-Trains can be shorter and instead run more frequently without increasing expenditure for staff.

3-Service frequency can easily be adjusted to meet sudden unexpected demands..

4-Intruder detection systems can be more effective than humans in stopping trains if someone is on the tracks.

5-Financial savings in both energy and wear-and-tear costs because trains are driven to an optimum specification.

FUTURE SCOPE

This project will be useful for the metro trains. This is very low cost project that can be help full in operating the train without driver. One can operate the train from remote distance. It will also be beneficial for the developing countries decreasing the cost of labors. Also the station information and emergency will also be displayed to the passengers. It will going to increase the technology trends. This is just a scratch to start this project there is a lot of improvement can be done in this project

APPENDIX

CODING FOR PROJECT

```
#include

void delay(unsigned int);

void lcdcmd(unsigned char);

void lcddata(unsigned char);

void lcdready();

void welcome();

void obstacle();
```

```
void station();

void gate();

sfr ldata = 0xA0;

sbit rs = P3^7;

sbit rw = P3^6;

sbit en = P3^5;

sbit busy = P2^7;

sbit m11 = P0^0;

sbit m12 = P0^1;

sbit m21 = P0^2;

sbit m22 = P0^3;

sbit alarm = P1^0;

sbit led = P1^1;

void ad(void) interrupt 0

{
P0 = 0x00;

alarm = 0;
```

```
obstacle();

}
void main(void)

{
IT0 = 0; // Configure interrupt 0 for falling edge on /INT0 (P3. 2)

EX0 = 1; // Enable EX0 Interrupt

EA = 1; // Enable Global Interrupt Flag

P0 = 0x00;

alarm = 1;

led = 1;

m11 = m12 = m21 = m22 = 0;

lcdcmd(0x38);

lcdcmd(0x0E);

lcdcmd(0x01);

lcdcmd(0x06);

lcdcmd(0x81);

while(1)
```

```
{
welcome();

station();

lcddata(' 1');

led = 0;

gate();

led = 1;

m11 = 1;

delay(1000);

m11 = 0;

station();

lcddata(' 2');

led = 0;

gate();

led = 1;

m11 = 1;

delay(1000);
```

```
m11 = 0;

station();

lcddata(‘ 3’);

led = 0;

gate();

led = 1;

m11 = 1;

delay(1000);

m11 = 0;

station();

lcddata(‘ 4’);

led = 0;

gate();

led = 1;

m11 = 1;

delay(1000);

m11 = 0;
```

```
station();

lcddata(‘ 5’);

led = 0;

gate();

led = 1;

m11 = 1;

delay(1000);

m11 = 0;

station();

lcddata(‘ 6’);

led = 0;

gate();

led = 1;

m11 = 1;

delay(1000);

m11 = 0;

station();
```

```
lcddata(' 7');

led = 0;

gate();

led = 1;

m11 = 1;

delay(1000);

m11 = 0;

station();

lcddata(' 8');

led = 0;

gate();

led = 1;

m11 = 1;

delay(1000);

m11 = 0;

station();

lcddata(' 9');
```

```
led = 0;

gate();

led = 1;

m11 = 1;

delay(1000);

m11 = 0;

station();

lcddata(‘ 1’);

lcddata(‘ 0’);

led = 0 ;

gate();

led = 1;

}

}
void gate()

{
m21 = 1;

m22 = 0;
```

```
delay(200);

m21 = m22 = 0;

delay(300);

m21 = 0;

m22 = 1;

delay(200);

m21 = m22 = 0;

}
void welcome()

{
unsigned char proj1[] = " Metro Rail";

unsigned char z;

lcdcmd(0x01);

lcdcmd(0x83);

for(z = 0; z <= 9; z++)

{
lcddata(proj1[z]);
```

```
}
delay(200);

}
void obstacle()

{
unsigned char obs1[] = " Obstacle";

unsigned char obs2[] = " Detected";

unsigned char z;

lcdcmd(0x01);

lcdcmd(0x84);

for(z = 0; z <= 7; z++)

{
lcddata(obs1[z]);

}
delay(100);

lcdcmd(0xC4);

for(z = 0; z <= 7; z++)

{
lcddata(obs2[z]);
```

```
}
}
void station()

{
unsigned char st[] = " Station ";

unsigned char z;

lcdcmd(0x01);

lcdcmd(0x83);

for(z = 0; z <= 7; z++)

{
lcddata(st[z]);

}
}
void delay(unsigned int itime)

{
unsigned int i, j;

for(i= 0; i for(j= 0; j <10000; j++);

}
void lcdcmd(unsigned char value)
```

```
{
lcdready();

ldata = value;

rs = 0;

rw = 0;

en = 1;

delay(1);

en = 0;

return;

}
void lcddata(unsigned char value)

{
lcdready();

ldata = value;

rs = 1;

rw = 0;

en = 1;

delay(1);
```

```
en = 0;

return;

}
void lcdready()

{
busy = 1;

rs = 0;

rw = 1;

while(busy == 1)

{
en = 0;

delay(1);

en = 1;

}
return;

}
```