OFFICE BOT

ECE 511 PROJECT

GROUP 10

12/02/2014

DEEPAK KARNIK

HRUSHIKESH KULKARNI

AKRITI AGARWAL

Abstract:

This Project 'Office Bot' is a Grid Follower Robot based on Microcontroller MSP430F5529 used to send intra-office memos and small packages within a workplace. This Robot follows a black line drawn over a white surface. The sensors are used to sense the line. When light signal is incident on the white surface, it gets reflected; and if it falls on the black surface, it is not reflected, this principle is used to scan the lines for the robot. All the above functioning is controlled by the microcontroller. The microcontroller is used to control the motors. It gets the signals from the sensors and it drives the motors according to the sensor inputs. Two DC motors are used to navigate the robot. The basic principle used for grid following is detecting the intersection and taking a turn accordingly. When a junction is detected, we take a single wheel turn, thus aligning the robot on to the right angled path. A small amount of delay is included after each turn to prevent false input to be taken by robot while turning. The robot also detects objects in its path using an IR transmitter and receiver pair by the beep of the buzzer. The robot was finally a success and functional, although we faced few complexities during the implementation.

Table of Contents:

Table of figures:

Motivation: A 'Robot' is a device that automates our work. The real motivation of the project was to create something different from the simple line follower which just follows a single path. The idea was to give an extra capability to the robot to detect grids/junctions and act accordingly. Whenever a junction occurs, two sensors sense it and send the feedback signal to microcontroller, now the microcontroller examines it and decides where to turn according to a program. In this way we can steer the robot wherever we want in the given arena in a minimum time as compared to line follower robot. The purpose of our project was to design a robot that is useful in carrying small objects from one office to the other and make life simpler and people more lazy!

Solution:

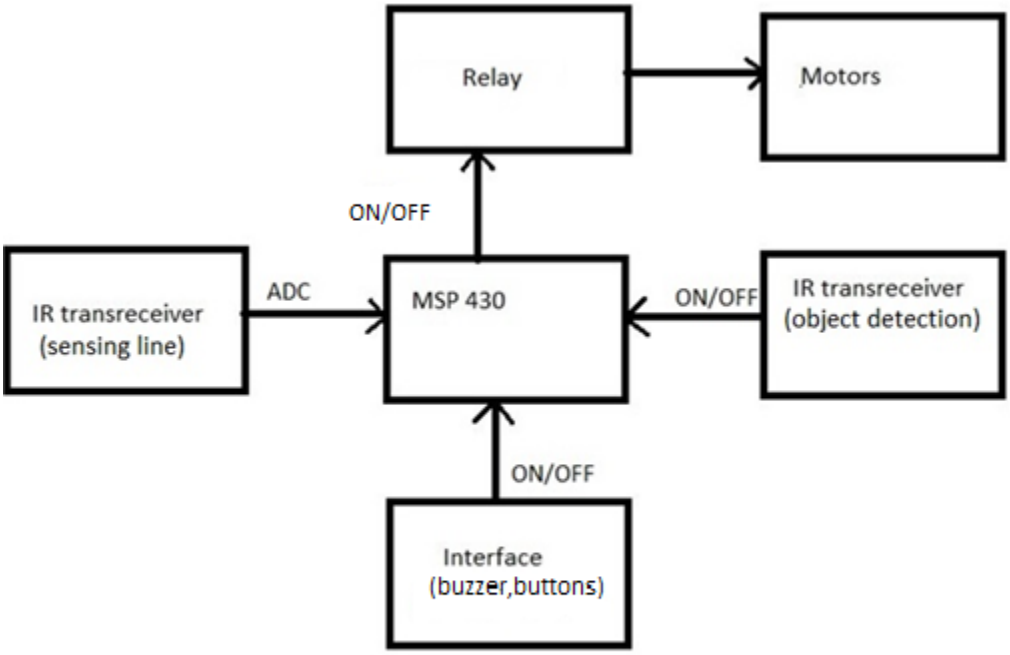

Fig.1 Block Diagram

The Office Bot uses a robust algorithm combined with IR sensors to detect black lines on a white surface and they transmit signal to the MSP430 via I/O port. If both the sensors detect black line at the same time, i.e. a junction, then it stops for a while (delay) and moves ahead or takes a right turn into the office depending on what it has been programmed to do. The turning radius is determined by stopping the right-hand side DC motor when the car is taking a right. The microcontroller receives the signal and actuates the motor via a relay. To determine when the robot needs to turn, i.e. which office it needs to enter is determined by a program, initiated by a push button - just like the working of an elevator.

When the IR transmitter-receiver pair detects any external body in the robot's way, it stops the movement of the robot and begins beeping a buzzer. As such, the software for each component was developed keeping in mind that it must coexist and share resources with other major components. Each of the major components are described in the sections that follow.

Hardware Interfacing:

1. IR LED Interfacing :

TSAL6200 is a high efficiency infrared emitting diode, molded in clear, blue grey tinted plastic packages. These emitters are ideally suitable as high performance replacements of standard emitters. In our project, IR LED TSAL6200 is interfaced with MSP 430 through pin P3.4. IR LED transmits IR pulses with 38 KHz frequency using timer and interrupt. To generate pulse of 38 KHz frequency, timer A is used to produce an interrupt 26 ms with 50% duty cycle. If an obstacle is present in front of the robot then IR pulses reflect back from an obstacle.

Fig.2 TSAL6200

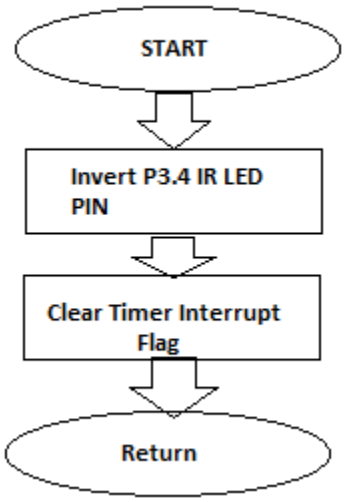

Fig.3 Timer Interrupt for IR led

2. IR Receiver Interfacing :

The IR receiver TSOP1738 is a photo-detector and a preamplifier in one package. It has high immunity against ambient light. TSOP 1738 detects on IR pulses with 38 KHz frequency. Three TSOP 1738 are interfaced with MSP430 through pins VCC, GND, and P3.3. TSOP1738 detects the reflected pulses from obstacle and gives a low output to MSP 430.

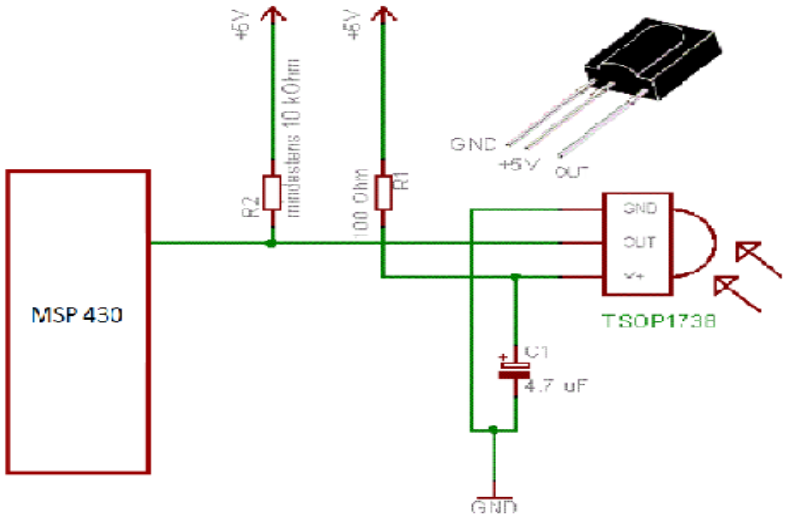

Fig 4. TSOP 1738 interfacing with MSP430

3. Line Sensor QRE1113:

The QRE1113 is a common reflectance sensor often used in robotic line followers. The sensor works by shining an IR LED down and seeing how much of that light bounces back using a phototransistor. Because dark colors will bounce back less of the light, the sensor can be used to tell the difference between white and black areas. So an array of these can be used to help a robot determine where a dark line is on the ground so it can follow it.

In our project, the sensors are interfaced via the in-built 12 bit ADC. The VCC and GND of sensor are connected to the on chip 3V and GND pins. The output pins of the sensors are connected to the ADC channel port 6. The left sensor is connected to the pin 6.0 and right sensor is connected to pin 6.2. Reference voltage used for ADC conversion is Vcc.

Fig.5 Line Sensor QRE1113

4. Solid state relay - Panasonic AQY212GH

A solid state relay is a static switch, which actuates a high-power circuit, upon receiving a small control voltage. It offers similar characteristics to a regular electromechanical relay, with added advantages such as a longer lifespan, contact-less switching, perfectly silent operation and a speedier performance. In the project, a pair of solid state relays is used to interface the two DC motors. Additionally, a buzzer, which operates on a 6V supply, is also interfaced to the MSP430 using a solid state relay.

Given below in the internal diagram of the device:

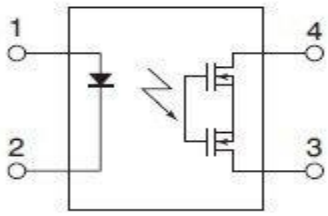

Fig.6 Schematic of AQY212GH

Here, pins 1 and 2 are connected to the control circuit, and pins 3 and 4 are connected to the load circuit. The input circuit is connected to a small LED, which transmits packets of light. These light packets are received by the switching device at the load side, (in this case, the switching device being a MOSFET), and the device completes the load circuit. The input and output are physically isolated, and the package is impervious to outside light, which is why the solid state relay is known to contain an optoisolator.

The primary use of the relays was to isolate the low power MSP430 microcontroller from components which work on a comparatively higher power.

Specifications:

- Switching device: photoMOS

- Continuous load current : 1.1A

- Control voltage range: 1.5V -5V

- Load voltage rating: 60V

- T_{ON}: 1.3ms

- T_{OFF}: 0.1ms

- I/O isolation voltage : 5000V

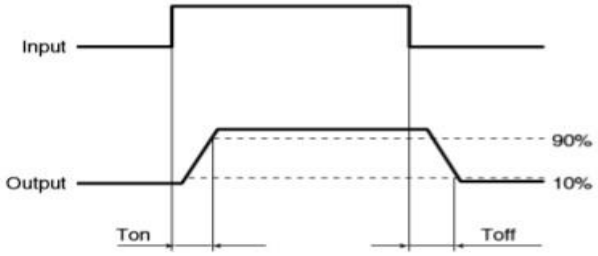

Fig.7 Timing diagram of the AQY212GH

5. Geared DC Motor:

Geared DC motors were coupled with wheels of car. It was used to drive car in the forward direction. Interfacing of DC motor with MSP 430 was done through the solid state relay (AQY212GH).

Specification:

- $6V_{DC}$

- 160 mA

- 100 RPM.

6. Piezo Buzzer:

A simple piezo buzzer will issue a (usually loud) buzz when it is powered. In our project, the buzzer has been interfaced with MSP430 as digital I/O and has been used for the purpose if obstacle detection .The buzzer beeps as soon as an obstacle is detected in its way.

7. Push Buttons :

In our project, we have used the two on-chip push buttons which are at pins 1.1 and 2.1 .The button at port 2.1 is used to start the robot and the other button would determine in which room the robot has to go.

Software interfacing:

```
// ADC AND MOTOR CONTROL

  P6SEL = 0x0F;                                          // Enable A/D channel inputs
  P2DIR = 0x05;  //P2.0,P2.2 Motor output pins
  ADC12CTL0 = ADC12ON+ADC12MSC+ADC12SHT0_8;   // Turn on ADC12, extend sampling time
                                                         // to avoid overflow of results
  ADC12CTL1 = ADC12SHP+ADC12CONSEQ_2;              // Use sampling timer, repeated sequence
  ADC12MCTL0 = ADC12INCH_0;                          // ref+=AVcc, channel = A0
  ADC12MCTL1 = ADC12INCH_1;                         // ref+=AVcc, channel = A1
  ADC12MCTL2 = ADC12INCH_2;                         // ref+=AVcc, channel = A2

 If((P2IN&0x02)!=0x02)                                //Check if button at pin2.1 is pressed
{
  do{
  ADC12CTL0 |= ADC12SC;                               // Start convn - software trigger
  while (!(ADC12IFG & BIT2));

  if( ADC12MEM0 >= 0xBB8 )                           //Left motor line detection

P2OUT &= ~BIT0 ;
     else{
          P2OUT |= BIT0 ;
     }
if( ADC12MEM2 >= 0xBB8 )                                 //Right motor line detect

P2OUT &= ~BIT2 ;
      else{
           P2OUT |= BIT2 ;
          }

 if ( ADC12MEM0  >= 0xBB8 && ADC12MEM2 >= 0xBB8 ) //Grid Detection
          {
                    P2OUT &= ~BIT0 ;
                    P2OUT &= ~BIT2 ;
                     __delay_cycles(500000);
          }
  }while(1);
}
```

```
// TIMER INTERRUPT FOR 38KHZ IR FREQUENCY
#include <msp430.h>


int main(void)
{
 WDTCTL = WDTPW + WDTHOLD;                          // Stop WDT
  P3DIR = 0xff                                      //Output
  TA1CCTL0 = CCIE;                                  // CCR0 interrupt enabled
  TA1CCR0 = 13;                                     // Counter value for 38khz
  TA1CTL = TASSEL_2 + MC_1 + TACLR;                 // SMCLK, upmode, clear TAR
  __bis_SR_register(GIE);                           // Enter LPM3, enable interrupts
}
// TIMER1 A0 INTERRUPT SERVICE ROUTINE
#if defined(__TI_COMPILER_VERSION__) || defined(__IAR_SYSTEMS_ICC__)
#pragma vector=TIMER1_A0_VECTOR
__interrupt void TIMER1_A0_ISR(void)
#elif defined(__GNUC__)
void __attribute__ ((interrupt(TIMER1_A0_VECTOR))) TIMER1_A0_ISR (void)
#else
#error Compiler not supported!
#endif
{  P3OUT ^= BIT4;                                   // Toggle P3.4
}
// FUNCTION TO TURN RIGHT
void right()
{
//Left Motor ON right Motor OFF
P2OUT |= BIT0;
```

```
 P2OUT &= ~BIT2;
        __delay_cycles(450000);                    //Approx delay for 90Degree turn
}

//BUTTON INTERRUPT FOR TAKING SECOND RIGHT

int t;

 P1DIR |= 0x00;                                    // Set P1.1 to input direction

 P1IE |= 0x02;                                     // P1.1 interrupt enabled

 P1IES |= 0x02;                                    // P1.1 Hi/lo edge

 P1IFG &= ~0x02;                                   // P1.1 IFG cleared

 _BIS_SR(GIE);                                     // Enable interrupt

 // PORT 1 INTERRUPT SERVICE ROUTINE

  #pragma vector=PORT1_VECTOR

  __interrupt void Port_1(void)

  {

     t=2;

    P1IFG &= ~0x02;                                          // P1.4 IFG cleared

  }
```

Results and Conclusion:

The Office Bot was completed as per proposal. The project was planned to be executed in three phases starting with identifying and procuring the required components as per design to carrying out standalone testing of the individual components to the eventual interfacing of all the components with software. The opportunity to learn was immense as we came across several challenges along the path, before we finally crossed the line. Few highlights of these are the realization of the importance of using the correct port numbers. We realized that overall complexity of the code could be reduced by making suitable port choices. The navigation scheme used by us involved using two sensors, processing them using the microcontroller, and using the outputs to drive the motor. We could have used a greater number of sensors to build a more sensitive robot, but since this would not have enhanced the objectives we stated in our project proposal, we decided to build a simple yet efficient robot. The proximity sensing was challenging with need for accurate calibration to extract optimum performance through which we realized several workarounds to tackle issues pertaining to these. We faced challenges like determining the threshold value for differentiating between the black and white lines. Also, fine tuning the IR emitter to generate a 38 KHz pulse was a little challenging. Also, the sensor that was ordered in the beginning for the obstacle detection, the TSSP583P8, was not giving the desired output and hence we had to order another sensor TSOP1738, which had similar characteristics to the sensor ordered initially and gave desired results. When these challenges were faced, we succeeded in working around them with clever programming alternatives. In the process, the team gained valuable experience integrating electronics with a low-resource microcontroller and managing the software accordingly.

Bibliography :

- **Panasonic Industrial Devices AQY212GH** - http://www.mouser.com/ProductDetail/Panasonic-Industrial-Devices/AQY212GH/?qs=sGAEpiMZZMvNy%2fd%2fTAiTWe1fehkLpwQtvPlptUrH%2fxY%3d
- **Vishay Semiconductors TSAL6200** - http://www.mouser.com/ProductDetail/Vishay-Semiconductors/TSAL6200/?qs=GAjK1cmKyrlKVEUhArzG2A%3D%3D
- **Vishay Semiconductors TSOP1738** - http://www.mouser.com/ProductDetail/Vishay-Semiconductors/TSOP1738/?qs=4rkkKKSASjvWmTn7s%252bwLXA%3D%3D
- **MSP430 LaunchPad PushButton** - http://processors.wiki.ti.com/index.php/MSP430_LaunchPad_PushButton
- **Vishay Semiconductors TSSP58P38** - http://www.mouser.com/ProductDetail/Vishay/TSSP58P38/?qs=%2fha2pyFaduhxHTPehEigO7%252b5MW2HnWg9QrRJMEWPtOY%3d
- **Robot Chassis "Magician" Kit** - http://www.amazon.com/Electronix-Express-ROBOT-CHASSIS-MAGICIAN/dp/B00GSCX4Z2/ref=sr_1_2?ie=UTF8&qid=1414984868&sr=8-2&keywords=magician+robot+chassis
- **SparkFun RedBot Sensor** - https://www.sparkfun.com/products/11769
- **RadioShack® 24VDC (6-28V) Piezo Buzzer** - http://www.radioshack.com/radioshack-24vdc-6-28v-piezo-buzzer/2730796.html#start=11
- **John H. Davies, "MSP430 Microcontroller Basics"**, Newnes, 2008, ISBN: 978-0-7506-8276-3
- **http://homepages.ius.edu/RWISMAN/C335/HTML/msp430Interrupts.HTM**
- **http://www.embeddedrelated.com/showarticle/182.php**
- **Demo codes from code composer studio for ADC and Timer.**

Appendix A: Team members Tasks

Deepak Karnik : Motor interface using ON/OFF, relays, electrical configuration, assembly and report writing

Hrushikesh Kulkarni : LDR/IR-sensors for line detection, ADC interface, assembly, software integration and report writing

Akriti Agarwal : IR-transmitter and receiver for buzzer, Push buttons/LEDs interface , software integration, assembly and report writing

Appendix B: Parts list

Component	Type	Quantity
MSP 430 Launch pad		1
TSOP 1738	IR sensor	1
Chassis		1
Batteries	1.5V AA	6
Piezo Buzzer		1
AQY212GH	Solid state relay	3
TSAL6200	IR LED	1
QRE1113	IR transceiver module	2
LED		2
DC motor		2

Appendix C: Schematic

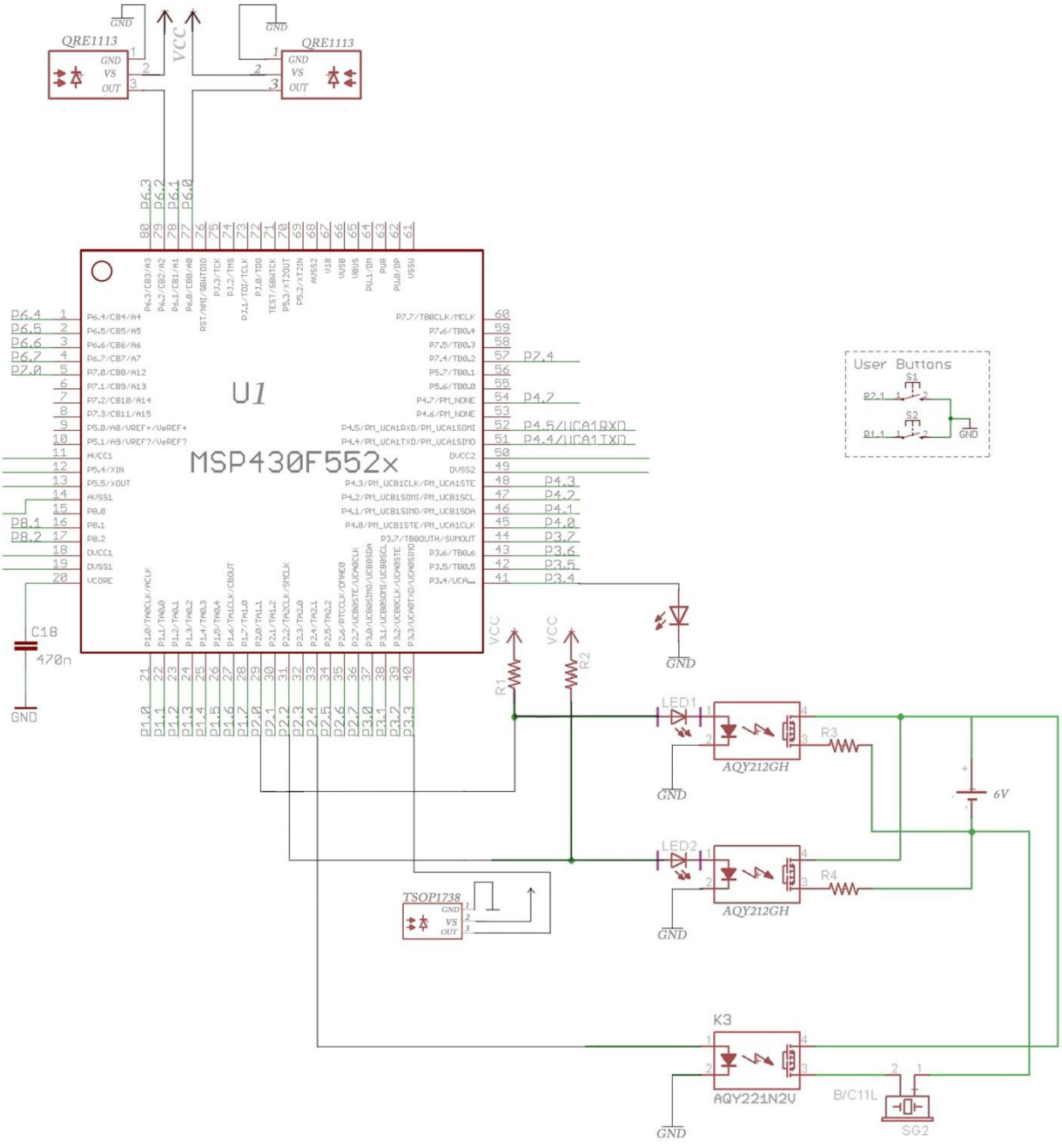

Fig.8 Schematic Diagram