



# SAFETY SYSTEM

TEAM 2

Ahmed Gamal

Amr Abdelsalam

# PHASE 1



# CODE

```
main.c E:\B\Workshops\Advanced Robotics K-vector\Final Project\Phase 1\phase1 final\phase1 final\main.c
/*
 * phase1 final.c
 *
 * Created: 24-Feb-18 6:46:37 PM
 * Author : Amr
 */

#include <avr/io.h> //header to enable data flow control over pins
#define F_CPU 1000000 //telling controller crystal frequency attached
#include <util/delay.h> //header to enable delay function in program

//THE KEYPAD FUNCTION
#define KeyDR DDRA //Keypad DirectionRegister(determines whether port pins will be used for input or output)
#define KeyCtrl PORTA //KeypadControl to output data on to port pins
#define KeypadVal PINA //KeypadVal to read data from input pins (port in)
char keypadScan()
{
    if(KeypadVal == 0xf0)
    {
        return 'z';
    }

    _delay_ms(50);
}
```

```
main.c E:\B\Workshops\Advanced Robotics K-vector\Final Project\Phase 1\phase1 final\phase1 final\main.c Go
uint8_t keypressedCode=KeypadVal; //uint8_t means unsigned char //max value 255
KeyDR ^= 0xff;
KeyCtrl ^= 0xff;
asm volatile("nop"); //do nothing
asm volatile("nop");
keypressedCode |=KeypadVal;
_delay_ms(50);

char x;
if(keypressedCode == 0b11101110)
x='1';
else if(keypressedCode == 0b11011110)
x='2';
else if(keypressedCode == 0b10111110)
x='3';
else if(keypressedCode == 0b01111110)
x='A';
else if(keypressedCode == 0b11101101)
x='4';
else if(keypressedCode == 0b11011101)
x='5';
else if(keypressedCode == 0b10111101)
x='6';
```



main.c X phase1 final

→ keypadScan

→ char keypadScan()

Go

```
else if(keypressedCode == 0b10111101)
x='6';
else if(keypressedCode == 0b01111101)
x='B';
else if(keypressedCode == 0b11101011)
x='7';
else if(keypressedCode == 0b11011011)
x='8';
else if(keypressedCode == 0b10111011)
x='9';
else if(keypressedCode == 0b01111011)
x='c';
else if(keypressedCode == 0b11100111)
x='*';
else if(keypressedCode == 0b11010111)
x='0';
else if(keypressedCode == 0b10110111)
x='#';
else if(keypressedCode == 0b01110111)
x='D';
else
{
    x='z';
}
```

100 %

```
main.c  X phase1 final
→ keypadScan  char keypadScan()  Go
if (x != 'z')
{
    DDRB = 0b00000010;
    PORTB |= 0b00000010;
    _delay_ms(500);
    PORTB &= 0b11111101;
}
return x;
}

//BLINKING FUNCTION

void blink ()
{
    PORTB |= 0b00000001;
    _delay_ms(500);
    PORTB &= 0b11111110;
    _delay_ms(500);
}

void blink_2()
{
    for (int i=0 ; i<=3 ; i=i+1)
    {
```

```
main.c  X  phase1 final
main.c  E:\B\Workshops\Advanced Robotics K-vector\Final Project\Phase 1\phase1 final\phase1 final\main.c  Go
PORTB |= 0b00000001;
_delay_ms(500);
PORTB &= 0b11111110;
_delay_ms(500);
}
}

//THE MAIN
int main (void)
{
    KeyDR=0x0f;    //taking column pins as output and rows pins as input
    KeyCtrl=0xf0;  //powering the column //pull up for 'f' in 0xf0

    DDRB = 0b00000001; //put the led in the first pin in port B
    const char password[4]= {'1','2','3','4'}; //the password we will set
    char data[4]; //put the data that comes from keypad in this array
    char test='0';
    int times=0;

    //////////////////////////////////////

    //THE LOOP
```

main.c X phase1 final

main.whi.le.for for (int i=0 ; i<=3 ; i=i+1)

```
//THE LOOP
```

```
while (1)
```

```
{char x;
```

```
////////// initializing the DATA ARRAY //////////
```

```
for (int i=0 ; i<4 ;i=i+1)
```

```
{ x=keypadScan();
```

```
while (x=='z')
```

```
{
```

```
    _delay_ms(50);
```

```
    x=keypadScan();
```

```
}
```

```
data[i]=x;
```

```
}
```

```
////////// compare with password //////////
```

```
for (int i=0 ; i<=3 ; i=i+1)
```

```
{
```

```
    if (data[i]==password[i])
```

```
    {
```

```
        test='1';
```

```
    }
```

```
else
```

100 %



main.c ✖ phase1 final

➔ main.while.for ▼ ⏮ for (int i=0 ; i<=3 ; i=i+1)

```
        else
        {
            test='0';
        }
    }

    ////////////////////////////////// ACTION //////////////////////////////////
    if (test=='1')
    {
        blink_2();
        times=0;
    }
    else
    {
        blink();
        times=times+1;
    }

    if (times==5)
    {
        _delay_ms(30000);
    }
}
```

100 % ◀

# FACED PROBLEMS AND SOLUTIONS

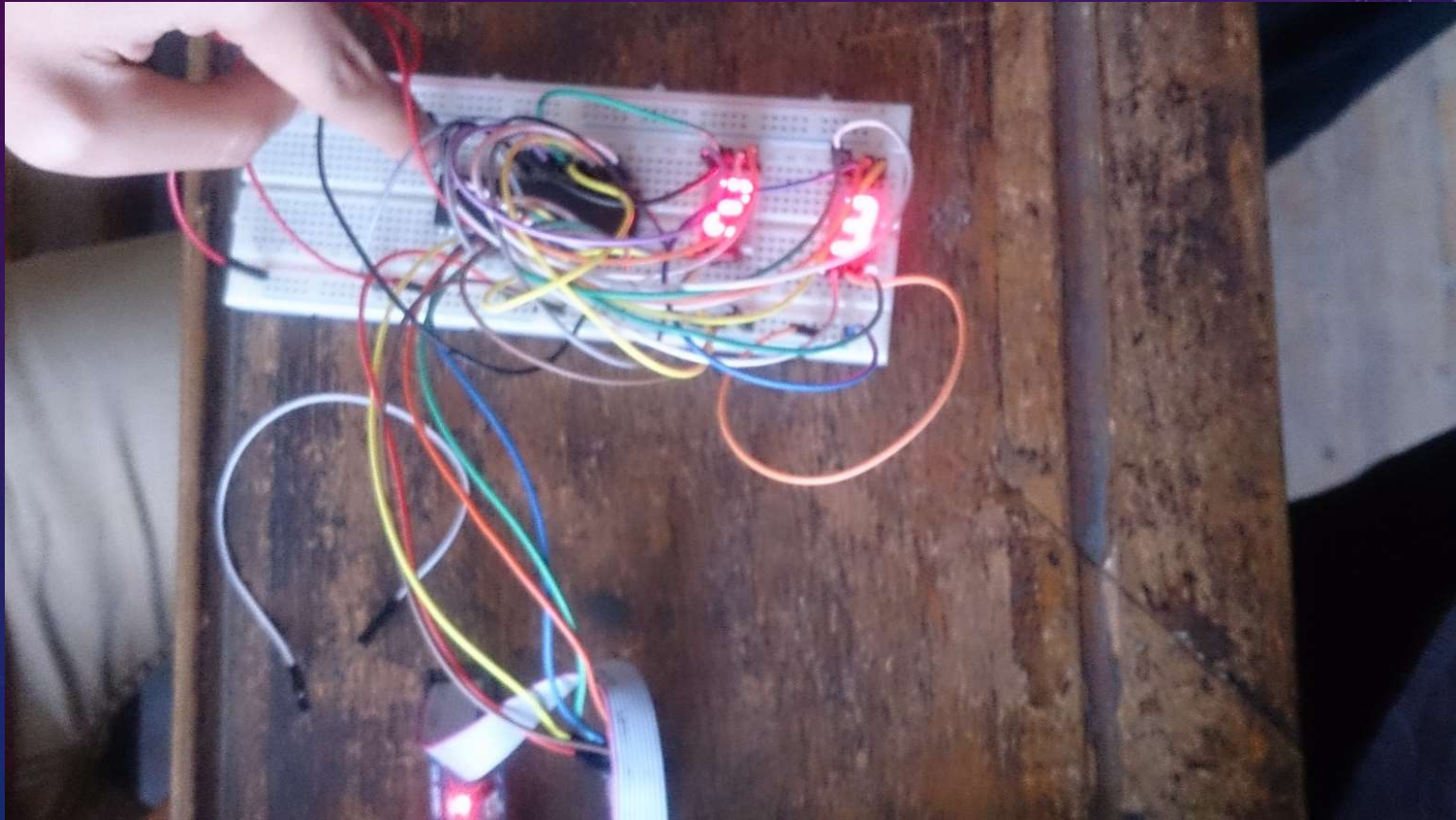
- Problem

Gathering the team to work.

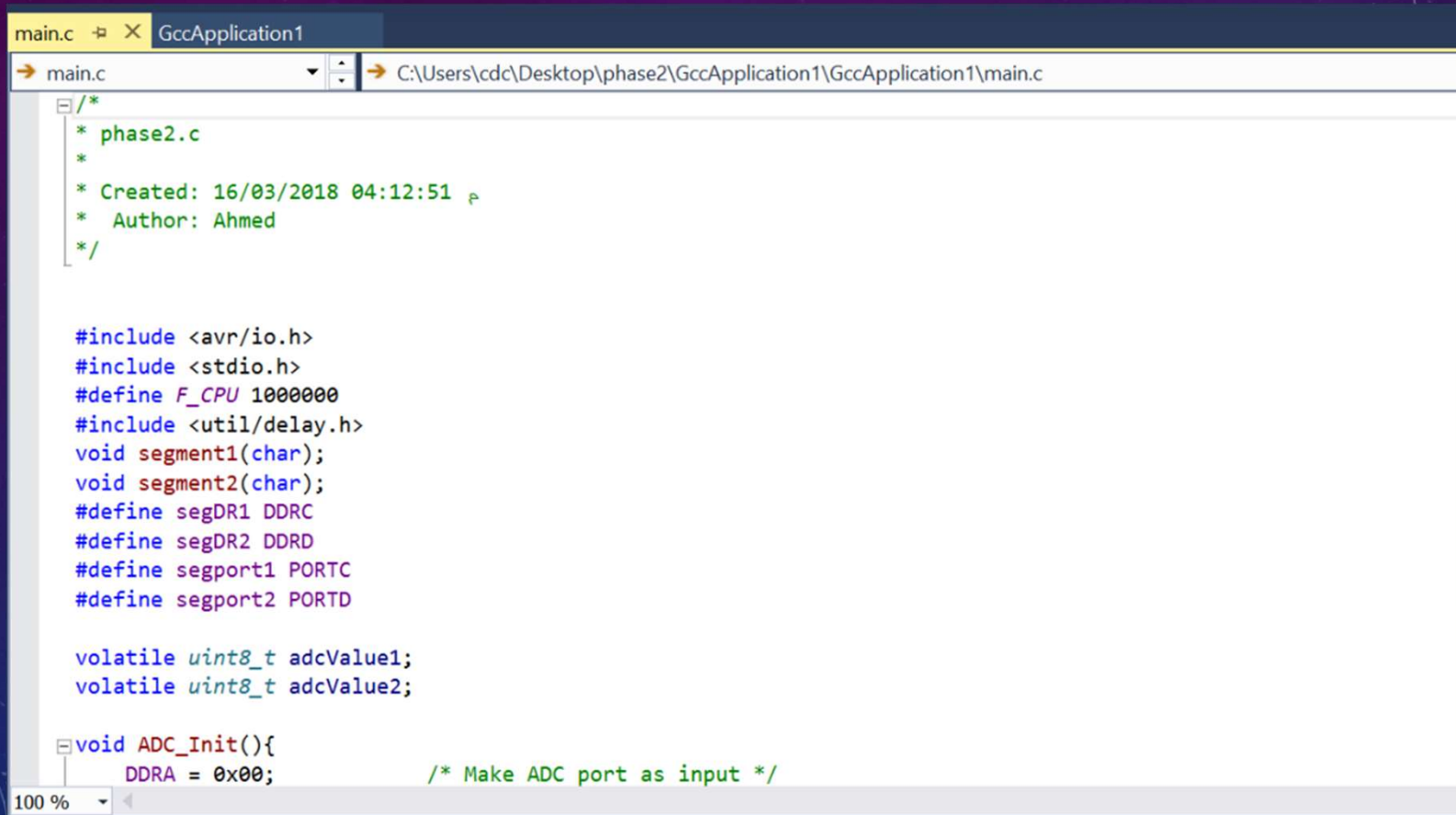
- Solution

We met on the delivery day on the coffee and finished it.

## PHASE 2



# CODE



```
main.c  X  GccApplication1
main.c  C:\Users\cdc\Desktop\phase2\GccApplication1\GccApplication1\main.c

/*
 * phase2.c
 *
 * Created: 16/03/2018 04:12:51
 * Author: Ahmed
 */

#include <avr/io.h>
#include <stdio.h>
#define F_CPU 1000000
#include <util/delay.h>
void segment1(char);
void segment2(char);
#define segDR1 DDRC
#define segDR2 DDRD
#define segport1 PORTC
#define segport2 PORTD

volatile uint8_t adcValue1;
volatile uint8_t adcValue2;

void ADC_Init(){
    DDRA = 0x00;          /* Make ADC port as input */
}
```



```
main.c X GccApplication1
main int main(void)
void ADC_Init(){
    DDRA = 0x00; /* Make ADC port as input */
    ADCSRA = 0x87; /* Enable ADC, with freq/128 */
    ADMUX = 0x40; /* Vref: Avcc, ADC channel: 0 */
}

int ADC_Read(char channel)
{
    ADMUX = 0x40 | (channel & 0x07); /* set input channel to read */
    ADCSRA |= (1<<ADSC); /* Start ADC conversion */
    while (!(ADCSRA & (1<<ADIF))); /* Wait until end of conversion by polling ADC interrupt flag */
    ADCSRA |= (1<<ADIF); /* Clear interrupt flag */
    _delay_ms(1); /* Wait a little bit */
    return ADCW; /* Return ADC word */
}

int main(void)
{
    //MCUCSR=(1<<7);
    MCUCSR=(1<<JTD);
    MCUCSR=(1<<JTD);
    segDR1=0x7f; //for 7 segment1
    segDR2=0x7f; //for 7 segment1
    char buffer[3];
}
```



main.c\* X GccApplication1

→ segment1.if ▾ else if (temp==)

```
ADC_Init();
int x=0;
while(1)
{
    x = (ADC_Read(0)*4.88);
    x = (x/10.00);

    int tempi =(int) x;//int temperature value

    snprintf(buffer, 3, "%d", tempi);
    segment1(buffer[0]);
    segment2(buffer[1]);

}
}
void segment1(char temp)
{
    if (temp=='0')
    {
        segport1=0x3f;
    }
    else if (temp=='1')
    {
        segport1=0x30;
    }
}
```

100 % ▾

main.c × GccApplication1

→ main.while ↕ while(1)

```
void segment1(char temp)
{
    if (temp=='0')
    {
        segport1=0x3f;
    }
    else if (temp=='1')
    {
        segport1=0x30;
    }
    else if (temp=='2')
    {
        segport1=0x5B;
    }
    else if (temp=='3')
    {
        segport1=0x4f;
    }
    else if (temp=='4')
    {
        segport1=0x66;
    }
    else if (temp=='5')
    {

```

100 %

main.c ✕ GccApplication1

→ main.while ▾ → while(1)

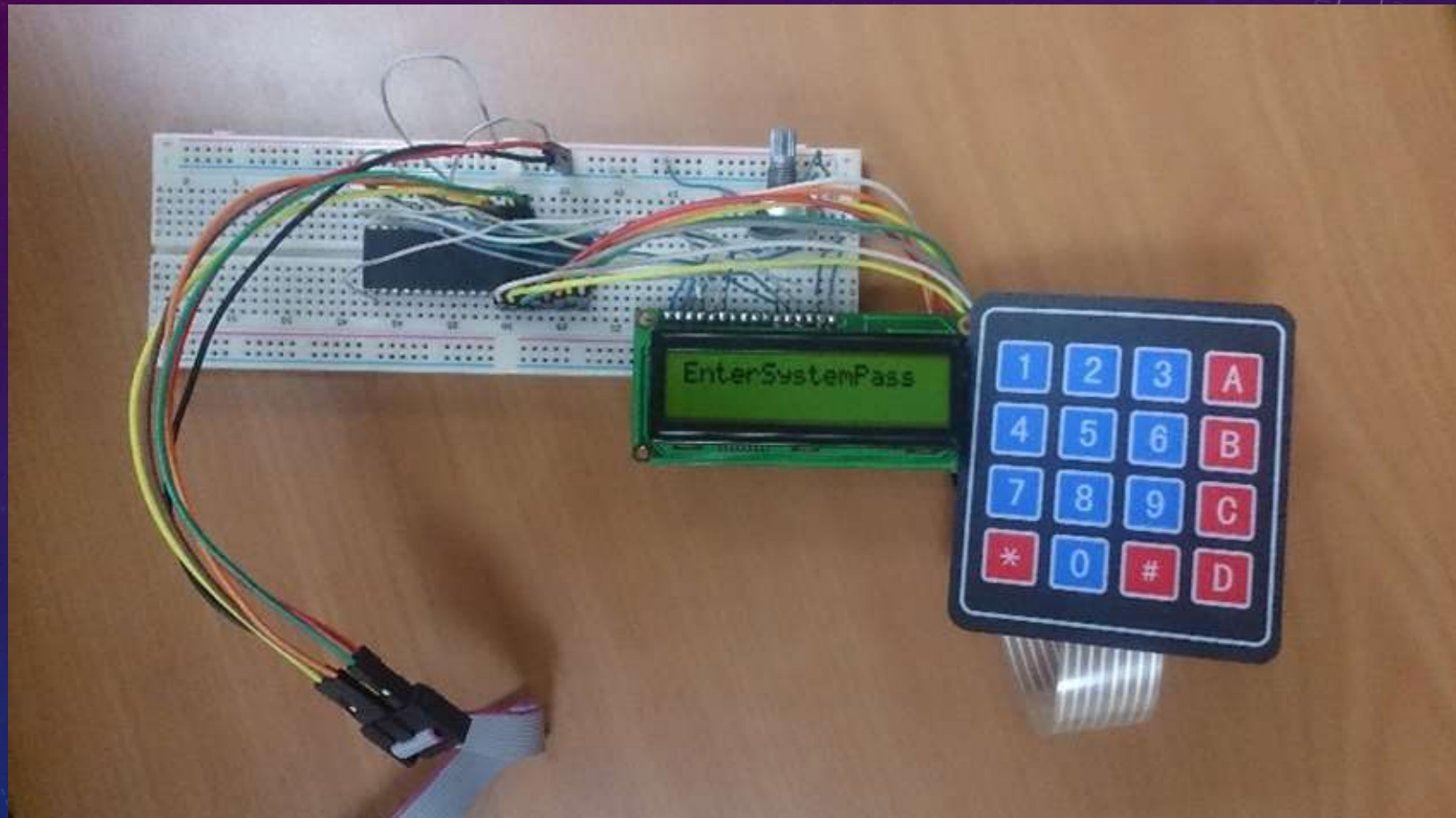
```
void segment2(char temp)
{
    if (temp=='0')
    {
        segport2=0x3f;
    }
    else if (temp=='1')
    {
        segport2=0x30;
    }
    else if (temp=='2')
    {
        segport2=0x5B;
    }
    else if (temp=='3')
    {
        segport2=0x4f;
    }
    else if (temp=='4')
    {
        segport2=0x66;
    }
    else if (temp=='5')
    {
```

100 % ▾

# FACED PROBLEMS AND SOLUTIONS

- Problem  
ADC was not working.
- Solution  
We had to use a code from the internet.

## PHASE 3





main.c X phase3.1

keypadScan char keypadScan(){...}

```
/*
 * phase3.1.c
 *
 * Created: 14-Apr-18 10:34:43 PM
 * Author : Amr
 */

#include <avr/io.h> //header to enable data flow control over pins
#define F_CPU 1000000 //telling controller crystal frequency attached
#include <util/delay.h> //header to enable delay function in program
#include <stdio.h>

//THE KEYPAD FUNCTION
#define KeyDR DDRA //Keypad DirectionRegister(determines whether port pins will be used for input or output)
#define KeyCtrl PORTA //KeypadControl to output data on to port pins
#define KeypadVal PINA //KeypadVal to read data from input pins (port in)
#define LCD_Dir DDRD /* Define LCD data port direction */
#define LCD_Port PORTD /* Define LCD data port */
#define RS PD0 /* Define Register Select pin */
#define EN PD1 /* Define Enable signal pin */

char keypadScan()...
```

100 %

main.c ✕ phase3.1

keypadScan char keypadScan(){...}

```
+ char keypadScan()...

+ void LCD_Command( unsigned char cmd )...

+ void LCD_Char( unsigned char data )...

+ void LCD_Init (void)          /* LCD Initialize function */...

+ void LCD_String (char *str)    /* Send string to LCD function */...

+ void LCD_String_xy (char row, char pos, char *str) /* Send string to LCD with xy position */...

+ void LCD_Clear()...

//THE MAIN

+ int main (void)...
```

100 %

# FACED PROBLEMS AND SOLUTIONS

- Problem
- No problem

# PHASE 4

- Live





*That's all Folks!*

*Any Question?*