אורט בראודה
המכללה האקדמית להנדסה

המחלקה להנדסת חשמל ואלקטרוניקה

פרויקט אמצע מיקרו בקרים

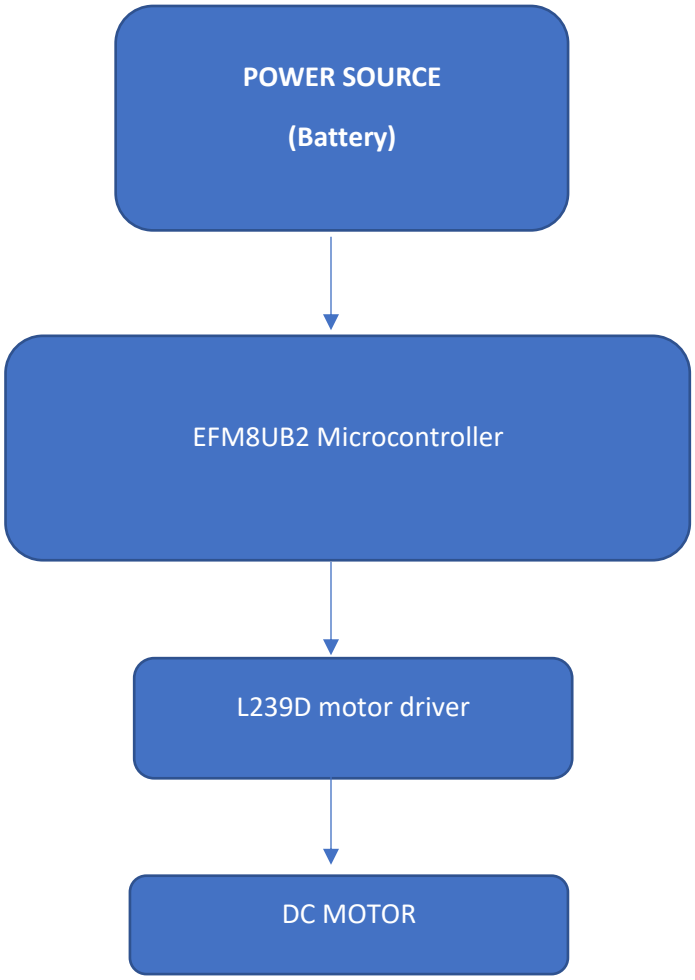
הפעלת מנוע **DC**

מחמוד חגה 318396355

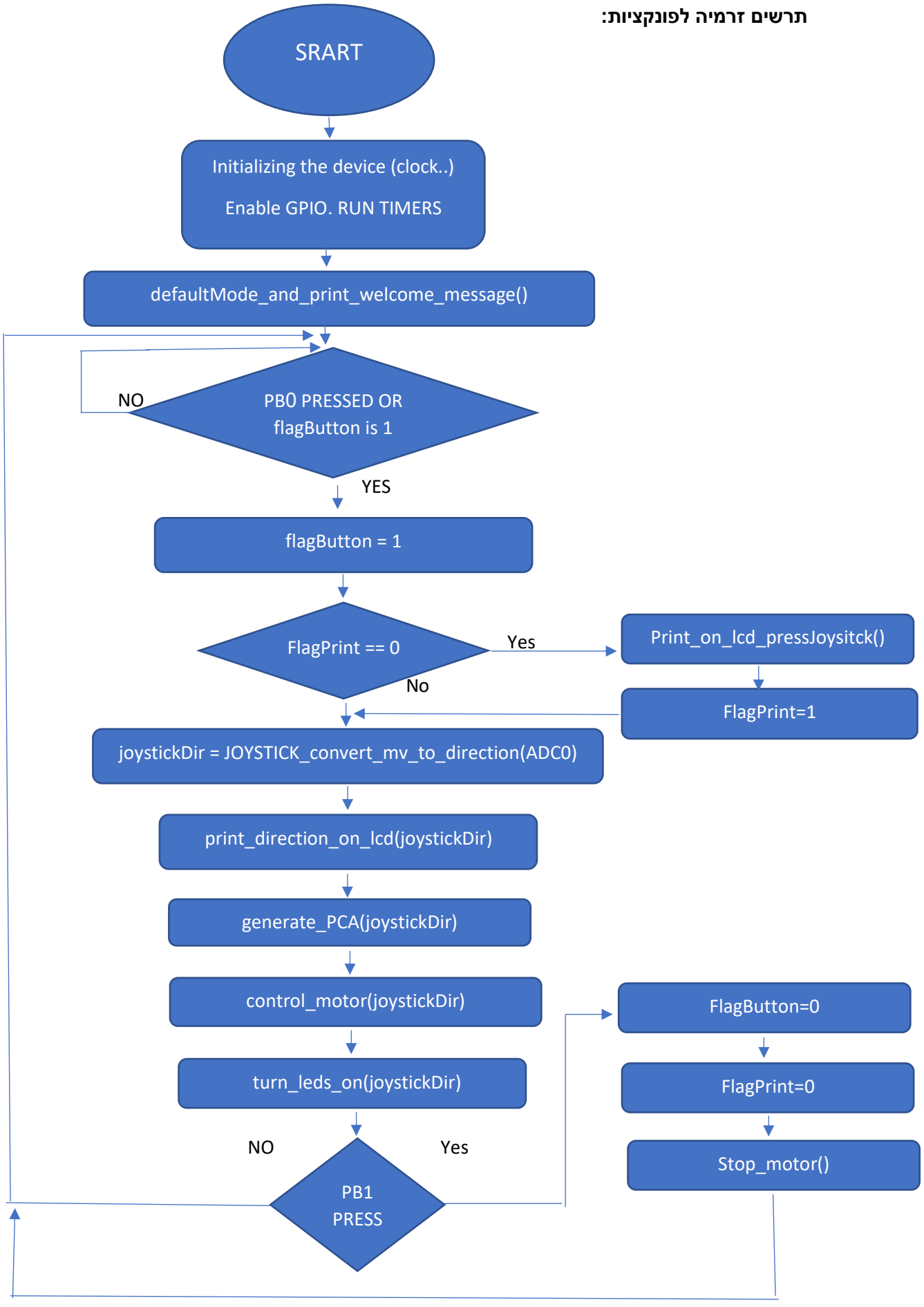
נאיל חסון 318386364

תוכן עניינים:

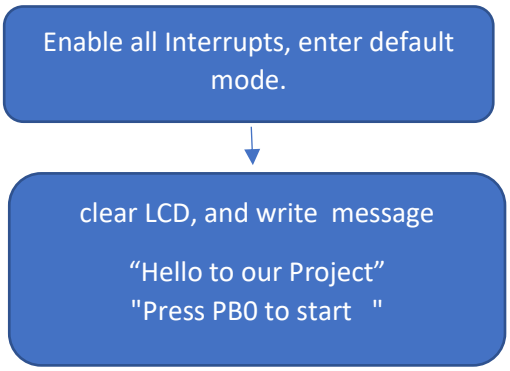
תרשים מלבנים למערכת:



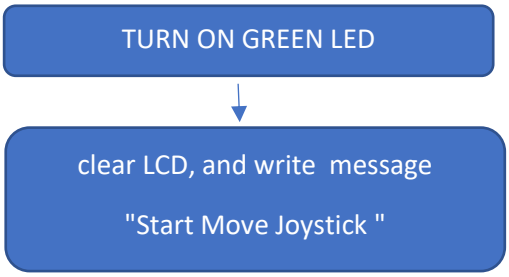
תרשים זרמיה לפונקציות:



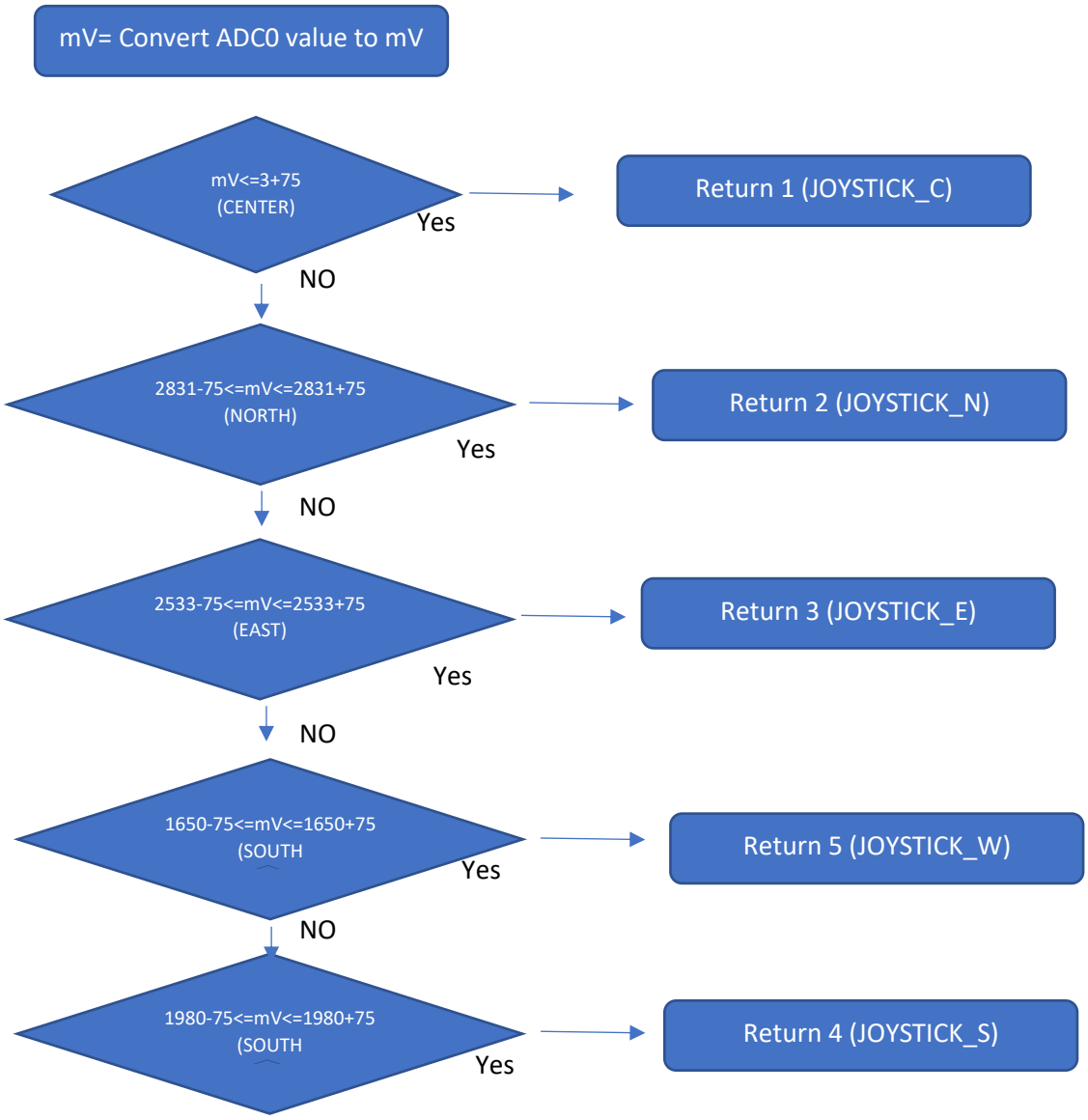
*** **defaultMode_and_print_welcome_message()**



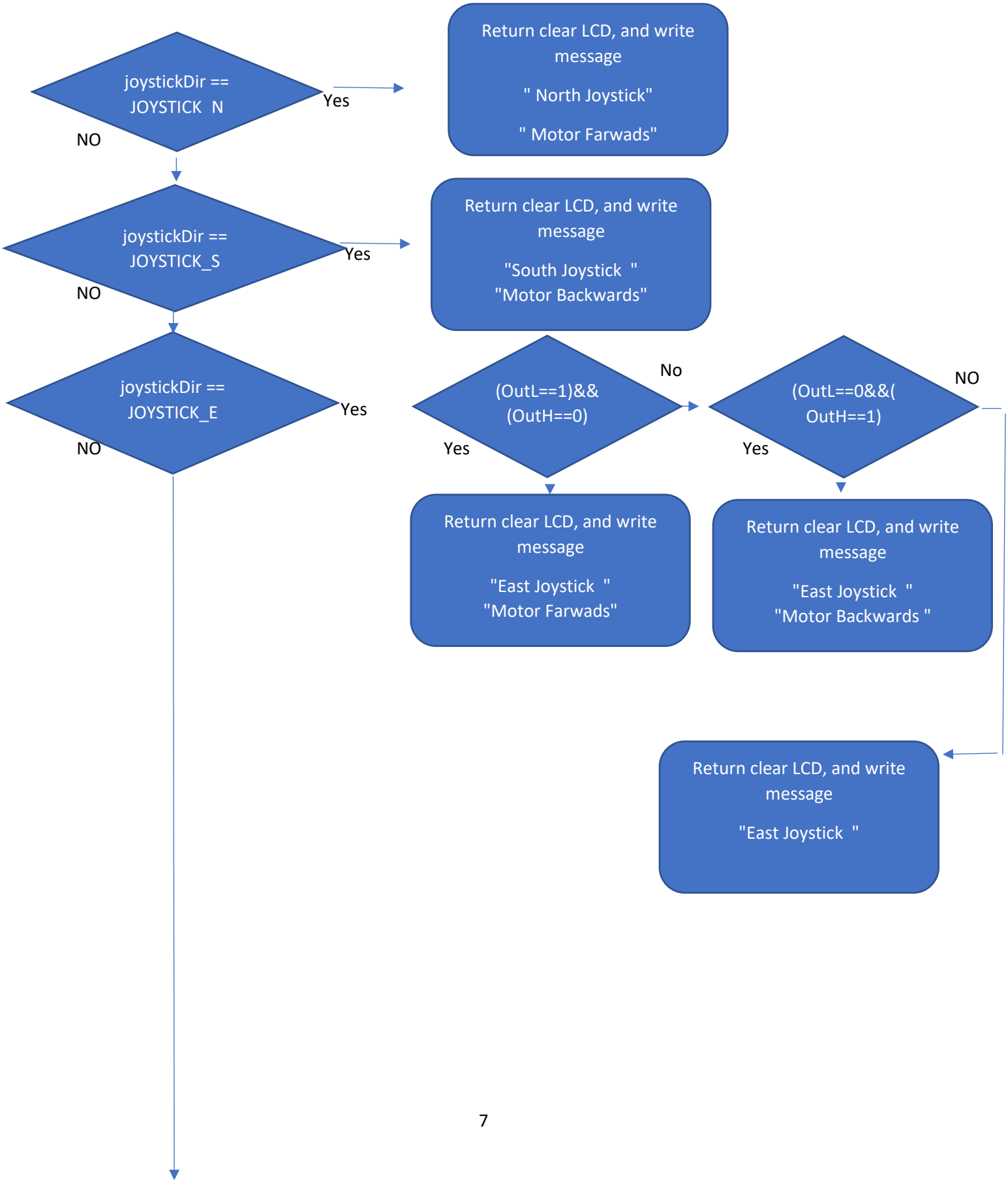
*** **print_on_lcd_pressJoystick()**

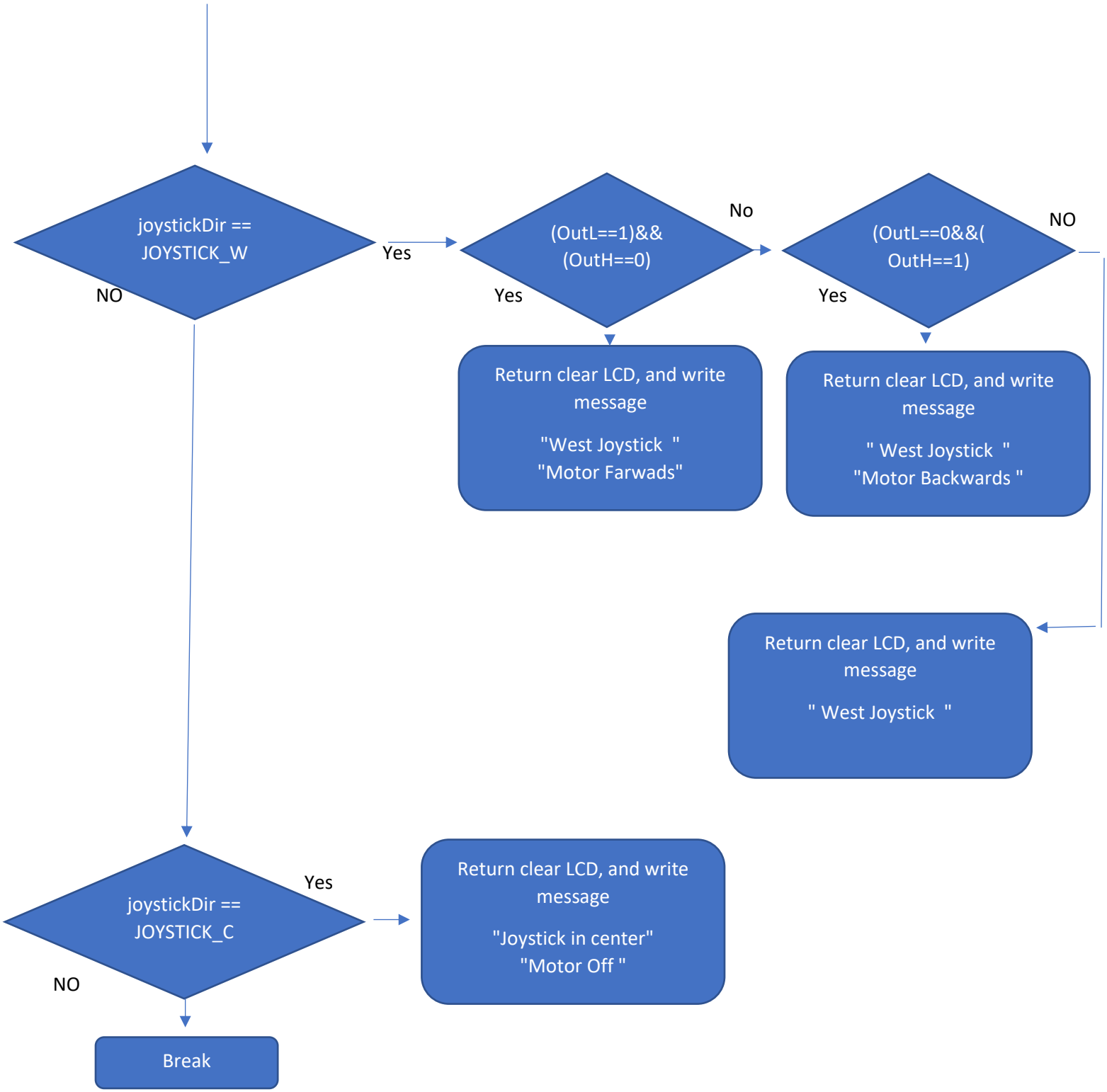


***** JOYSTICK_convert_mv_to_direction(ADC0)**

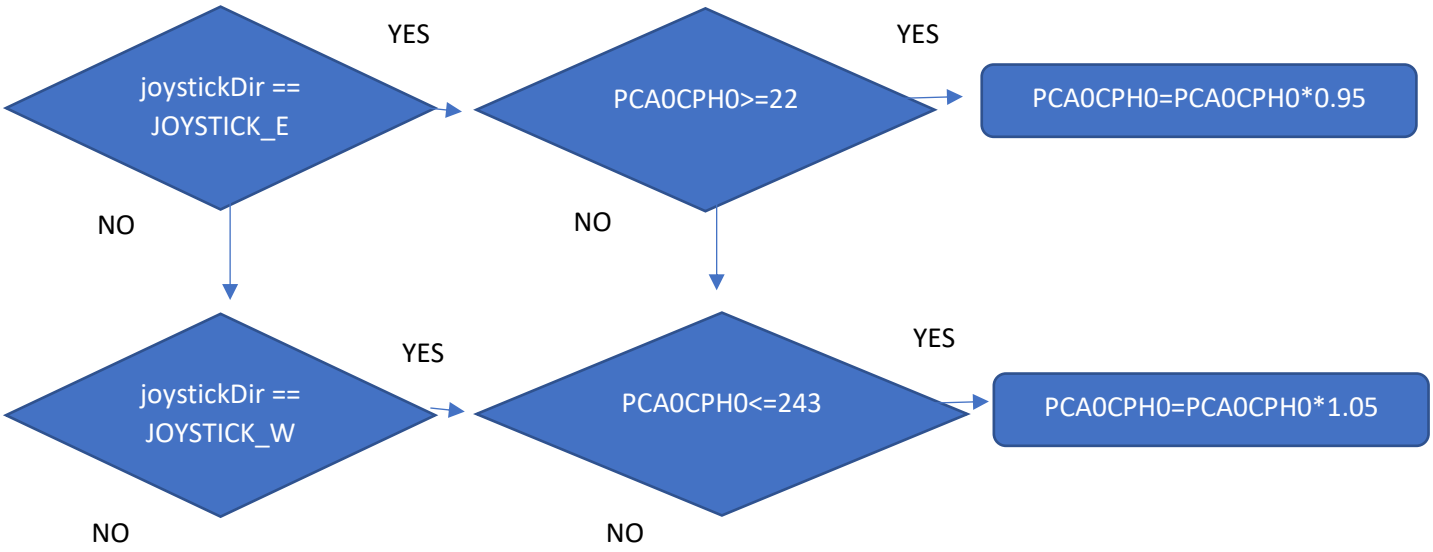


***** print_direction_on_lcd (joystickDir)**

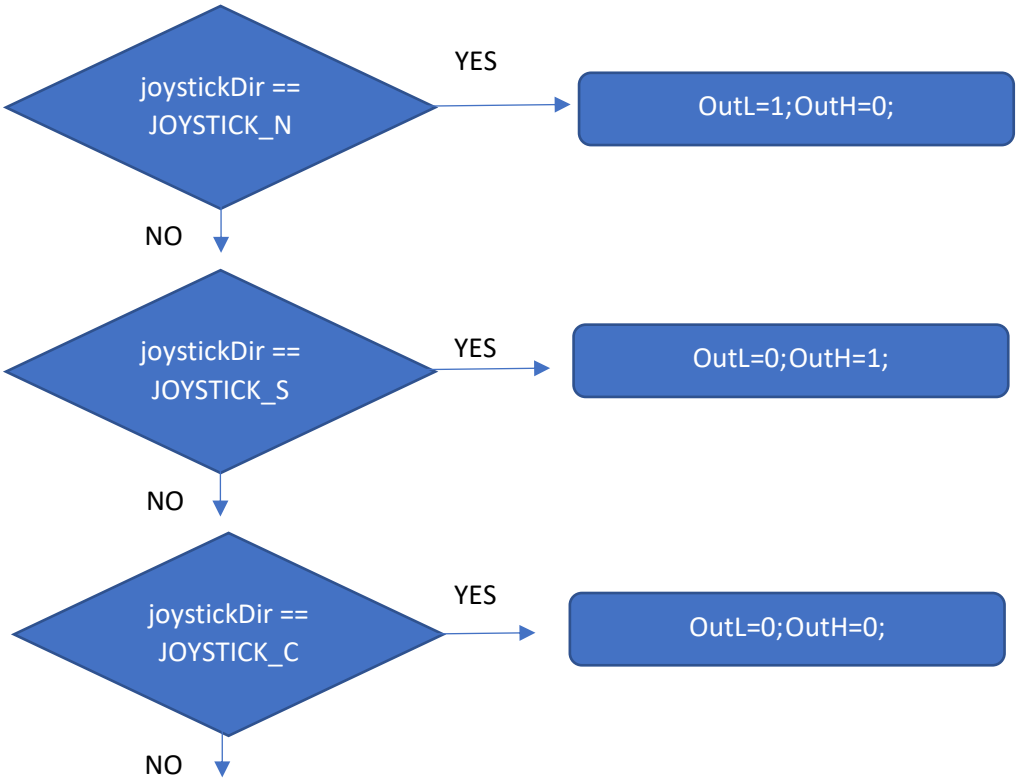



joystickDir == JOYSTICK_W
Yes
NO
(OutL==1)&& (OutH==0)
No
Yes
(OutL==0&&(OutH==1)
NO
Yes
Return clear LCD, and write message
"West Joystick "
"Motor Farwads"
Return clear LCD, and write message
" West Joystick "
"Motor Backwards "
Return clear LCD, and write message
" West Joystick "
joystickDir == JOYSTICK_C
Yes
NO
Return clear LCD, and write message
"Joystick in center"
"Motor Off "
Break

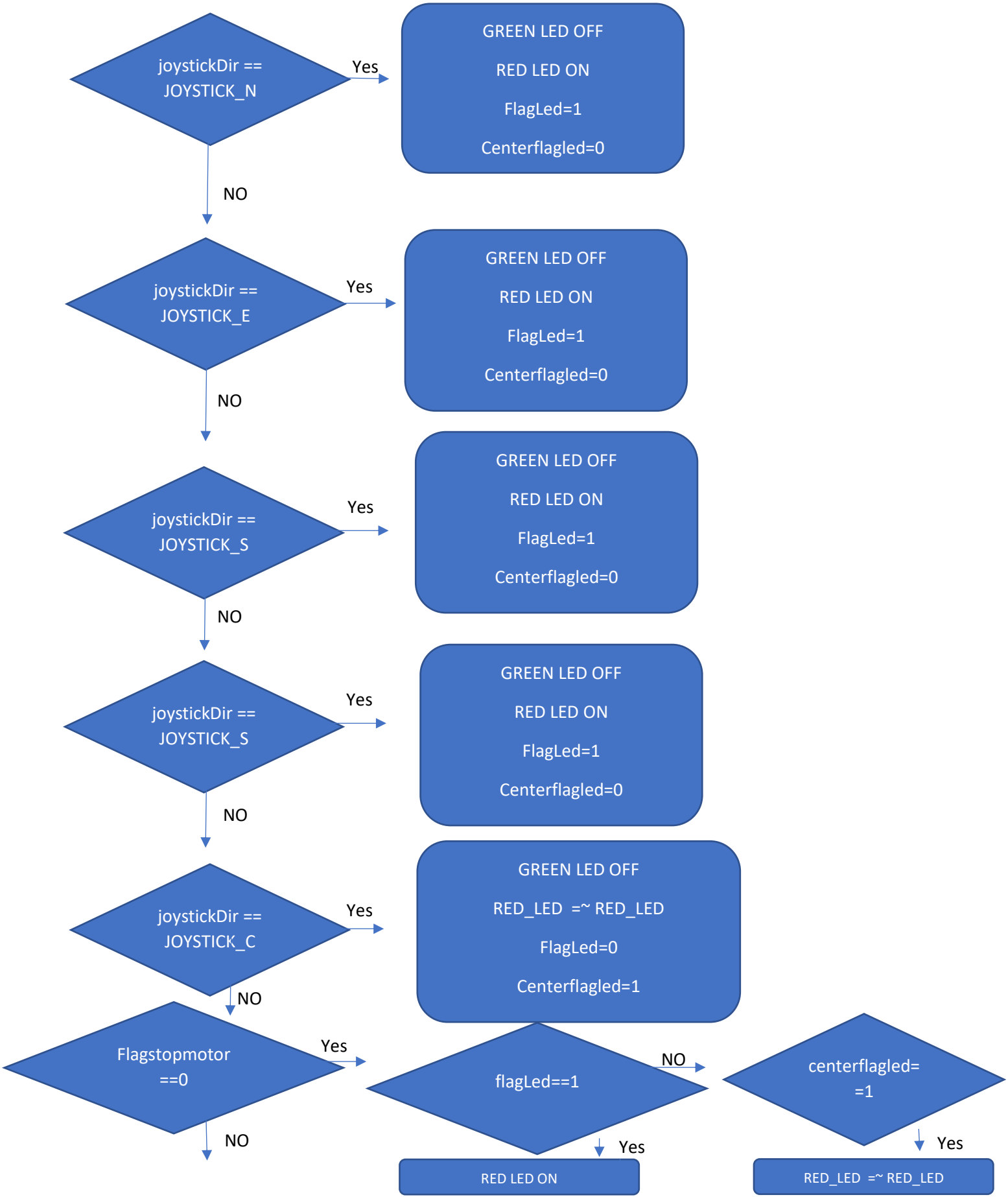
***** generate_PCA(joystickDir)**



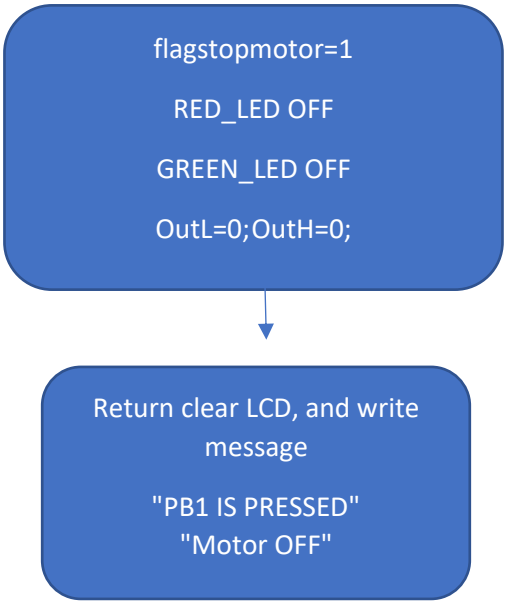
***** control_motor (joystickDir)**



***** turn_leds_on(joystickDir)**



***** Stop_motor()**


flagstopmotor=1
RED_LED OFF
GREEN_LED OFF
OutL=0;OutH=0;
Return clear LCD, and write message
"PB1 IS PRESSED"
"Motor OFF"

PROGRAM CODE:

MAIN

```
#include "bsp.h"
#include "InitDevice.h"
#include "disp.h"
#include "tick.h"
#include "render.h"
#include "adc_0.h"
#include "joystick.h"
#include "Headfile.h"
#include <SI_EFM8UB2_Register_Enums.h>
sbit pb0=P0^2; //buton0
sbit pb1=P0^3; //button1
int flagButton=0;//flag to check when pb0 is pressed, so the program run
int flagprint=0; // when pb0 is pressed, we print a message just one time.
void SiLabs_Startup (void){}// Disable the watchdog here
int main(void)
{
    defaultMode_and_print_welcome_message();// default and print welcome message
      while (1)
      {
          if(pb0==0||flagButton==1)
          {
              flagButton=1;
              if(flagprint==0)
              {
                  print_on_lcd_pressJoystick();// print open message to move the joystick
                  flagprint=1;
              }
              joystickDir = JOYSTICK_convert_mv_to_direction(ADC0); // Get joystick direction
              print_direction_on_lcd(joystickDir);//print the direction of the joystick on lcd
              generate_PCA(joystickDir);//generete pca (x 0.95%)||(x 1.05%)
              control_motor(joystickDir);//control the motor (farwards || backwards || stop)
              turn_leds_on(joystickDir);//leds ON || OFF
          }
          if(pb1==0)
          {
              flagButton=0;
              flagprint=0;
              Stop_motor();// stop the mottor and print a message
          }
      }
}
```

Header file:

```
#ifndef SRC_HEADFILE_H_
#define SRC_HEADFILE_H_
////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////
#define JOYSTICK_NONE          0       ///< not pressed
#define JOYSTICK_C             1       ///< center
#define JOYSTICK_N             2       ///< north
#define JOYSTICK_E             3       ///< east
#define JOYSTICK_S             4       ///< south
#define JOYSTICK_W             5       ///< west

#define JOYSTICK_MV_ERROR 75

#define JOYSTICK_MV_C          3       ///< center position in mV
#define JOYSTICK_MV_N          2831    ///< north position in mV
#define JOYSTICK_MV_E          2533    ///< east position in mV
#define JOYSTICK_MV_S          1650    ///< south position in mV
#define JOYSTICK_MV_W          1980    ///< west position in mV
////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////
#define LCD_height 128
SI_SBIT (DISP_EN, SFR_P2, 2);         // Display Enable
#define DISP_BC_DRIVEN   0            // 0 = Board Controller drives display
#define DISP_EFM8_DRIVEN 1            // 1 = EFM8 drives display
#define DISP_BC_DRIVEN   0            // 0 = Board Controller drives display
#define DISP_EFM8_DRIVEN 1            // 1 = EFM8 drives display
#define LED_ON  (0)
#define LED_OFF (1)
#define VREF_MV         (3300UL)
#define ADC_MAX_RESULT  ((1 << 10)-1) // 10 bit ADC

   uint8_t joystickDirection;
SI_SBIT (BC_EN, SFR_P1, 6);
SI_SBIT(GREEN_LED, SFR_P1, 6);         // Green LED
SI_SBIT(RED_LED, SFR_P2, 0);          // Red LED
sbit OutL = P2 ^ 6; // pin input in the motor L293d
sbit OutH = P2 ^ 7; // pin input in the motor L293d
SI_SEGMENT_VARIABLE(line[DISP_BUF_SIZE], uint8_t, RENDER_LINE_SEG);// draw on lcd
uint8_t y; // draw on lcd
uint16_t mV; // value of adc0 in mv
uint8_t joystickDir; // joystick direction (south, north, east, west)
int flagLed=0;  // check if the joystick is pressed in any direction, and not in the Center press.
int centerflagled=0;// check if the joystick in center, to blink the led
int flagstopmotor=0;
```

```
////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////
void defaultMode_and_print_welcome_message()// default and print welcome message
{
    enter_DefaultMode_from_RESET();  // Enter default mode
    DISP_EN = DISP_BC_DRIVEN;           // Display not driven by EFM8
    IE_EA = 1;  // Enable all interrupts
    DISP_Init(); // clear lcd
    for(y=0; y<FONT_HEIGHT;y++)
        {
          RENDER_StrLine(line,1,y,"Hello to our Project");
          DISP_WriteLine(50+y,line);    Wait(1);
          RENDER_StrLine(line,1,y,"Press PB0 to start         ");
          DISP_WriteLine(60+y,line);    Wait(1);
        }
}
////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////
void print_on_lcd_pressJoystick()// print open message to move the joystick
{
    GREEN_LED=LED_ON;
    DISP_Init(); // clear lcd
    for(y=0; y<FONT_HEIGHT;y++)
        {
          RENDER_StrLine(line,1,y,"Start Move Joystick ");
          DISP_WriteLine(50+y,line);    Wait(1);
        }
}
////////////////////////////////////////////////////////////////////////////////////
```

```
//////////////////////////////////////////////////////////////////////////////
uint8_t JOYSTICK_convert_mv_to_direction(uint16_t adc0)// Get joystick direction
{
    // The measured voltage applied to P2.5 is given by:
      //
      //                            Vref (mV)
      //   measurement (mV) =   --------------- * result (bits)
      //                        (2^10)-1 (bits)
    mV=((uint32_t)adc0 * VREF_MV) / ADC_MAX_RESULT;

        // determine which direction pad was pressed
        if ((mV <= JOYSTICK_MV_C + JOYSTICK_MV_ERROR))//CENTER
        {
            flagstopmotor=0;
            return JOYSTICK_C;
        }
        else if ((mV >= JOYSTICK_MV_N - JOYSTICK_MV_ERROR) && \
                 (mV <= JOYSTICK_MV_N + JOYSTICK_MV_ERROR))// NORTH
        {
            flagstopmotor=0;
            return JOYSTICK_N;
        }
        else if ((mV >= JOYSTICK_MV_E - JOYSTICK_MV_ERROR) && \
                 (mV <= JOYSTICK_MV_E + JOYSTICK_MV_ERROR))//EAST
        {
            flagstopmotor=0;
            return JOYSTICK_E;
        }
        else if ((mV >= JOYSTICK_MV_S - JOYSTICK_MV_ERROR) && \
                 (mV <= JOYSTICK_MV_S + JOYSTICK_MV_ERROR))//SOUTH
        {
            flagstopmotor=0;
            return JOYSTICK_S;
        }
        else if ((mV >= JOYSTICK_MV_W - JOYSTICK_MV_ERROR) && \
                 (mV <= JOYSTICK_MV_W + JOYSTICK_MV_ERROR))//WEST
        {
            flagstopmotor=0;
            return JOYSTICK_W;
        }
        else
        {
            return JOYSTICK_NONE;// NO DIRECTION
        }
}
//////////////////////////////////////////////////////////////////////////////
```

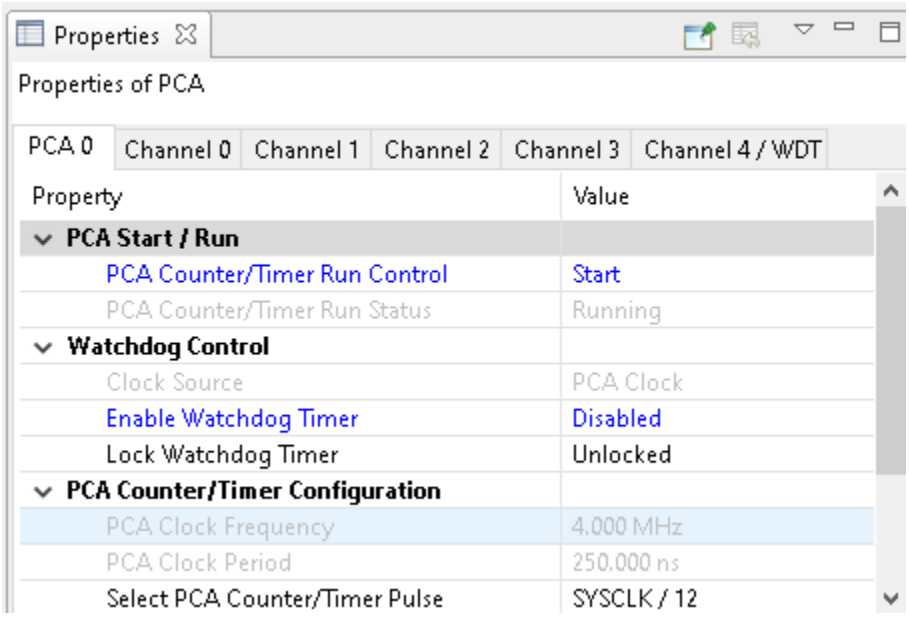
```
void print_direction_on_lcd(uint8_t joystickDir)//print the direction of the joystick on lcd
{

    switch (joystickDir)
      {
      case JOYSTICK_N:
        {
            DISP_Init();
            for(y=0; y<FONT_HEIGHT;y++)
                {
                  RENDER_StrLine(line,1,y,"North Joystick ");
                  DISP_WriteLine(50+y,line);    Wait(1);
                  RENDER_StrLine(line,1,y," Motor Farwads");
                  DISP_WriteLine(60+y,line);    Wait(1);

                }break;}
      case JOYSTICK_S:
        {
            DISP_Init();
            for(y=0; y<FONT_HEIGHT;y++)
            {
                RENDER_StrLine(line,1,y,"South Joystick  ");
                DISP_WriteLine(50+y,line);  Wait(1);
                RENDER_StrLine(line,1,y," Motor Backwards");
                DISP_WriteLine(60+y,line);  Wait(1);
            }break;}
      case JOYSTICK_E:
        {
            DISP_Init();
            if((OutL==1)&&(OutH==0)) //joystick North, motor Farwads
            {
                for(y=0; y<FONT_HEIGHT;y++)
                {
                    RENDER_StrLine(line,1,y,"East Joystick          ");
                    DISP_WriteLine(50+y,line);  Wait(1);
                    RENDER_StrLine(line,1,y," Motor Farwads");
                    DISP_WriteLine(60+y,line);  Wait(1);
                }
            }
            if((OutL==0)&&(OutH==1)) //joystick south, motor backwards
            {
                for(y=0; y<FONT_HEIGHT;y++)
                {
                    RENDER_StrLine(line,1,y,"East Joystick          ");
                    DISP_WriteLine(50+y,line);  Wait(1);
                    RENDER_StrLine(line,1,y," Motor Backwards");
                    DISP_WriteLine(60+y,line);  Wait(1);
                }
            }break;}
      case JOYSTICK_W:
        {
            DISP_Init();
            if((OutL==1)&&(OutH==0)) //joystick North, motor Farwads
            {
                for(y=0; y<FONT_HEIGHT;y++)
                    {
                        RENDER_StrLine(line,1,y,"West Joystick              ");
                        DISP_WriteLine(50+y,line);  Wait(1);
                        RENDER_StrLine(line,1,y," Motor Farwads");
                        DISP_WriteLine(60+y,line);  Wait(1);
                    }
            }
            if((OutL==0)&&(OutH==1)) //joystick south, motor backwards
            {
                for(y=0; y<FONT_HEIGHT;y++)
                    {
                        RENDER_StrLine(line,1,y,"West Joystick            ");
                        DISP_WriteLine(50+y,line);  Wait(1);
                        RENDER_StrLine(line,1,y," Motor Backwards");
                        DISP_WriteLine(60+y,line);  Wait(1);
                    }
            }break;}
      case JOYSTICK_C:
        {
            DISP_Init();
            for(y=0; y<FONT_HEIGHT;y++)
            {
                RENDER_StrLine(line,1,y,"Joystick in center.");
                DISP_WriteLine(50+y,line);  Wait(1);
                RENDER_StrLine(line,1,y," Motor Off        ");
                DISP_WriteLine(60+y,line);  Wait(1);
            }break;}
      }
}
```

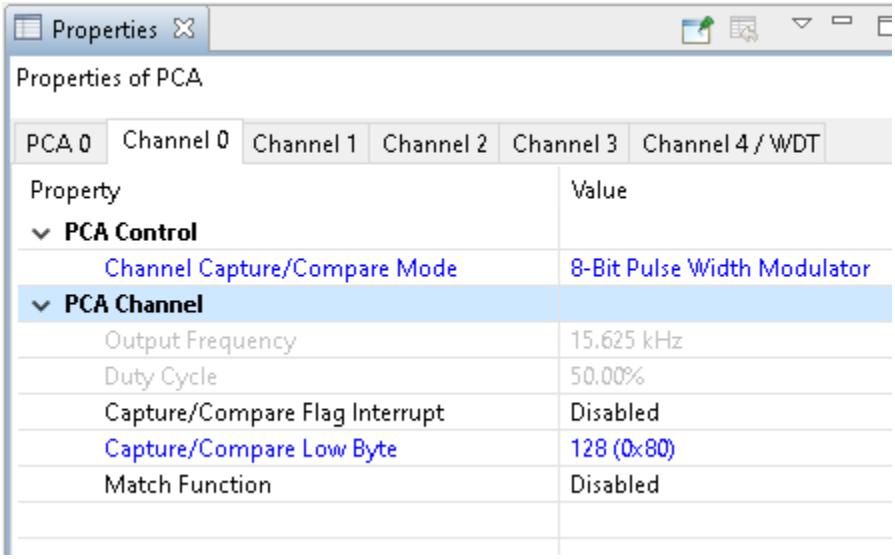
```
.....................................................................................................
void generate_PCA(uint8_t joystickDir)//generete pca (x 0.05%)||(x 1.05%)
{
    switch (joystickDir)
      {
      case JOYSTICK_E:
        {
            if(PCA0CPH0>=22)// we ask this so we dont highter the pwm to arrive to 100%. then we cant get it back and control it
                PCA0CPH0=PCA0CPH0*0.95;  //higher pwm
            break;}
      case JOYSTICK_W:
        {
            if(PCA0CPH0<=243)// we ask this so we dont lower the pwm to arrive to 0%. then we cant get it back and control it
                PCA0CPH0=PCA0CPH0*1.05; // lower pwm
            break;}
      }
}
////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////
void control_motor(uint8_t dir)//control the motor (farwards || backwards || stop)
{
      switch (dir)
      {
      case JOYSTICK_N:
      {
          OutL=1;   OutH=0; break;
      }
      case JOYSTICK_S:
      {
          OutL=0;   OutH=1; break;
      }
      case JOYSTICK_C:
      {
          OutL=0;   OutH=0; break;
      }
}
}
////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
void turn_leds_on(uint8_t joystickDir)//leds ON || OFF
{
      switch (joystickDir)
      {
      case JOYSTICK_N:
        {
            GREEN_LED=LED_OFF;
            flagLed=1;
            centerflagled=0;
            RED_LED  =LED_ON;
        break;}
      case JOYSTICK_E:
        {
            GREEN_LED=LED_OFF;
            RED_LED  =LED_ON;
            flagLed=1;
            centerflagled=0;
        break;}
      case JOYSTICK_S:
        {
            GREEN_LED=LED_OFF;
            RED_LED  =LED_ON;
            flagLed=1;
            centerflagled=0;
        break;}
      case JOYSTICK_W:
        {
            GREEN_LED=LED_OFF;
            RED_LED  = LED_ON;
            flagLed=1;
            centerflagled=0;
            break;}
      case JOYSTICK_C:
        {
            GREEN_LED=LED_OFF;
            RED_LED  =~ RED_LED;
            flagLed=0;
            centerflagled=1;
            break;}

      default:
          if(flagstopmotor==0)
          {
              if(flagLed==1)
                  RED_LED  =LED_ON;
              if(centerflagled==1)
                  RED_LED  =~ RED_LED;
          } break;}

}
```

```
void Stop_motor()// stop the mottor and print a message
{
    flagstopmotor=1;
    RED_LED  =LED_OFF;
    GREEN_LED=LED_OFF;
    OutL=0; OutH=0;
    DISP_Init();
    for(y=0; y<FONT_HEIGHT;y++)
    {
        RENDER_StrLine(line,1,y,"PB1 IS PRESSED");
        DISP_WriteLine(50+y,line);  Wait(1);
        RENDER_StrLine(line,1,y," Motor Off          ");
        DISP_WriteLine(60+y,line);  Wait(1);
    }
}
//////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////
#endif /* SRC_HEADFILE_H_ */
```

1. PCA:

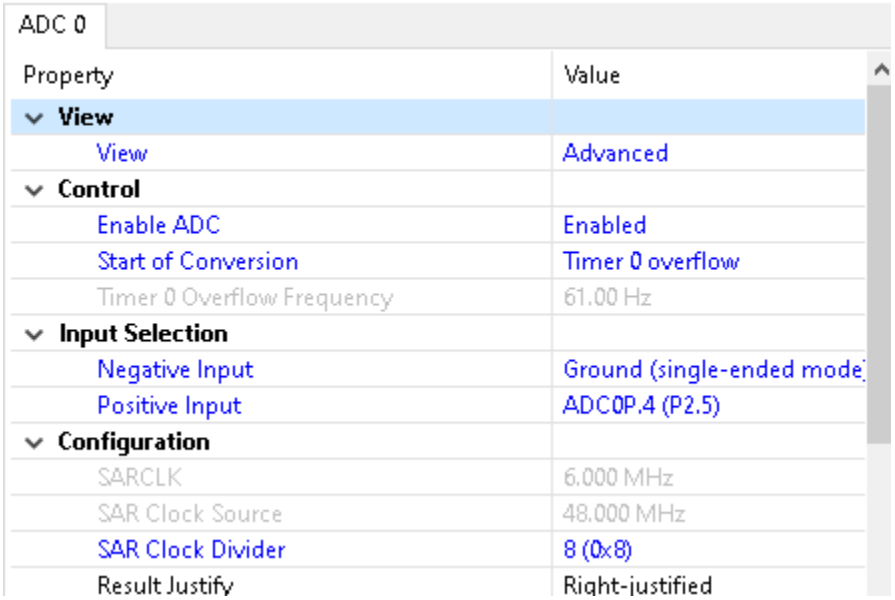




PCA for disabling Watchdog and generating PWM at channel0 of PCA0 in 50% duty cycle at the start of the run.

The value of capture/compare low byte is 128 as it gives as the 50% duty cycle.

2. ADC0:

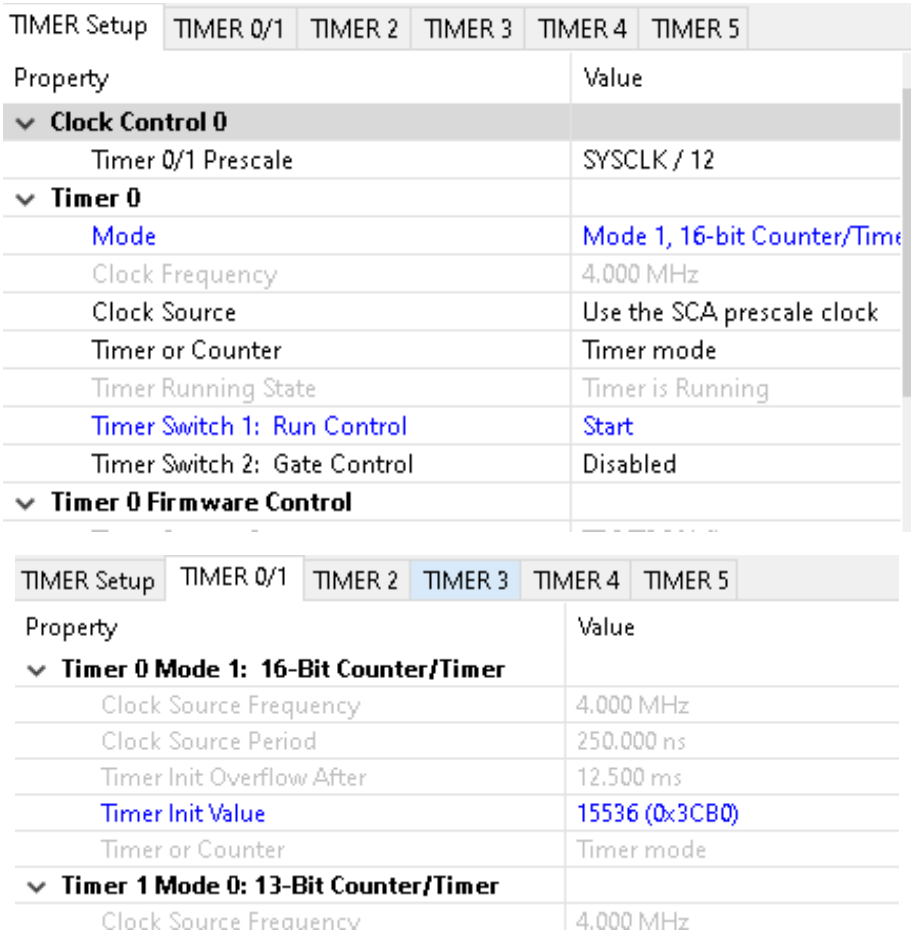


ADC0 is enabled for the joystick and we need to read every 50[ms] so we use TIMER0 for the time count.

The joystick connected to P2.5 so we put the positive input to P2.5.

3. Timers:

TIMER0:



TIMER0 is started to count the time for reading the joystick every 50[ms].

To find the init value for TIMER0 so reach overflow every 50[ms], we used this formula:

$$F_{TIMER0} = \frac{F_{input\ clock}}{2^{16} - TH0} \rightarrow TH0 = 2^{16} - \frac{F_{input\ clock}}{F_{TIMER0}} = 15536$$

$F_{input\ clock}$ = 1 [MHz]

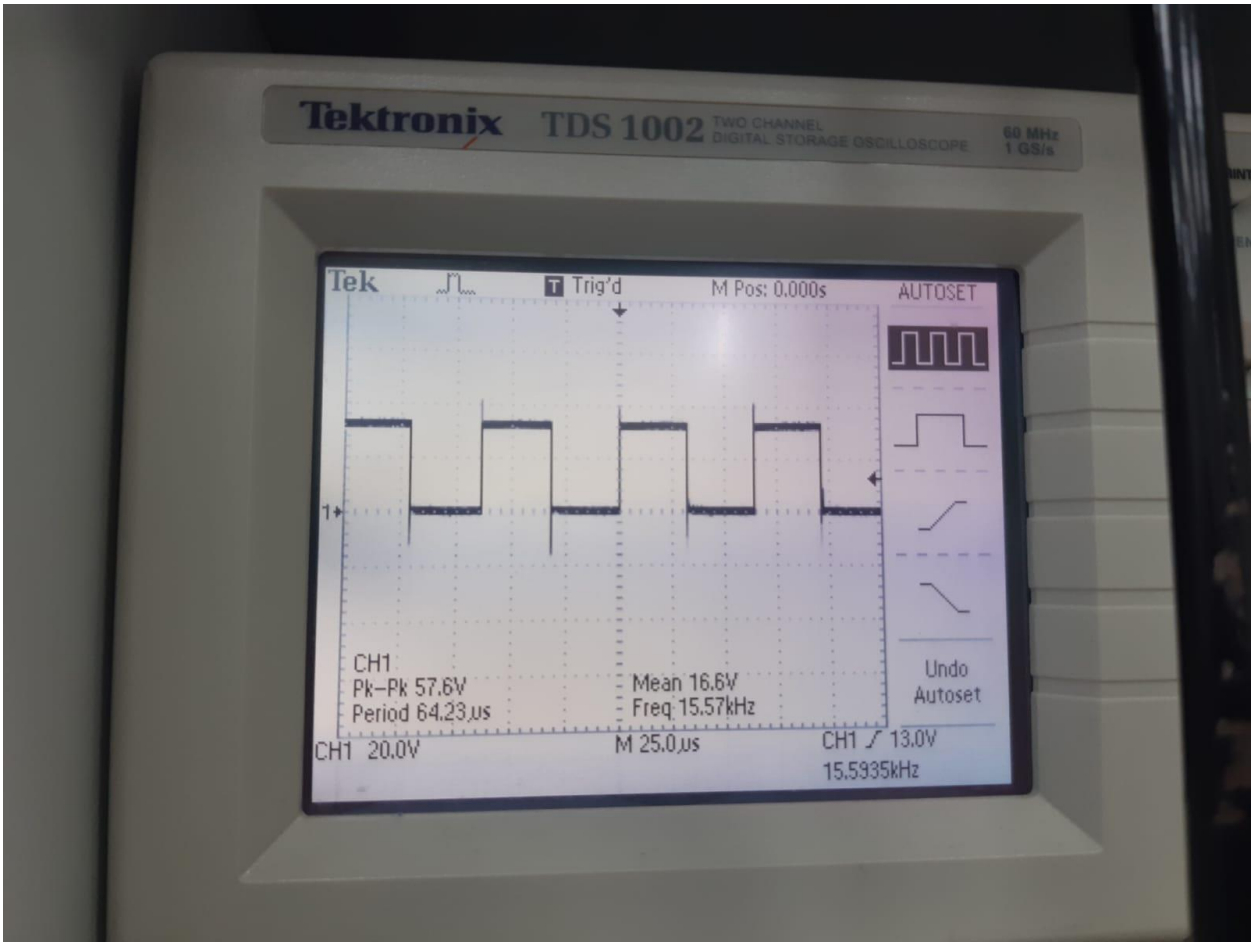
F_{TIMER0} = 20 [Hz]

4. Interrupts:

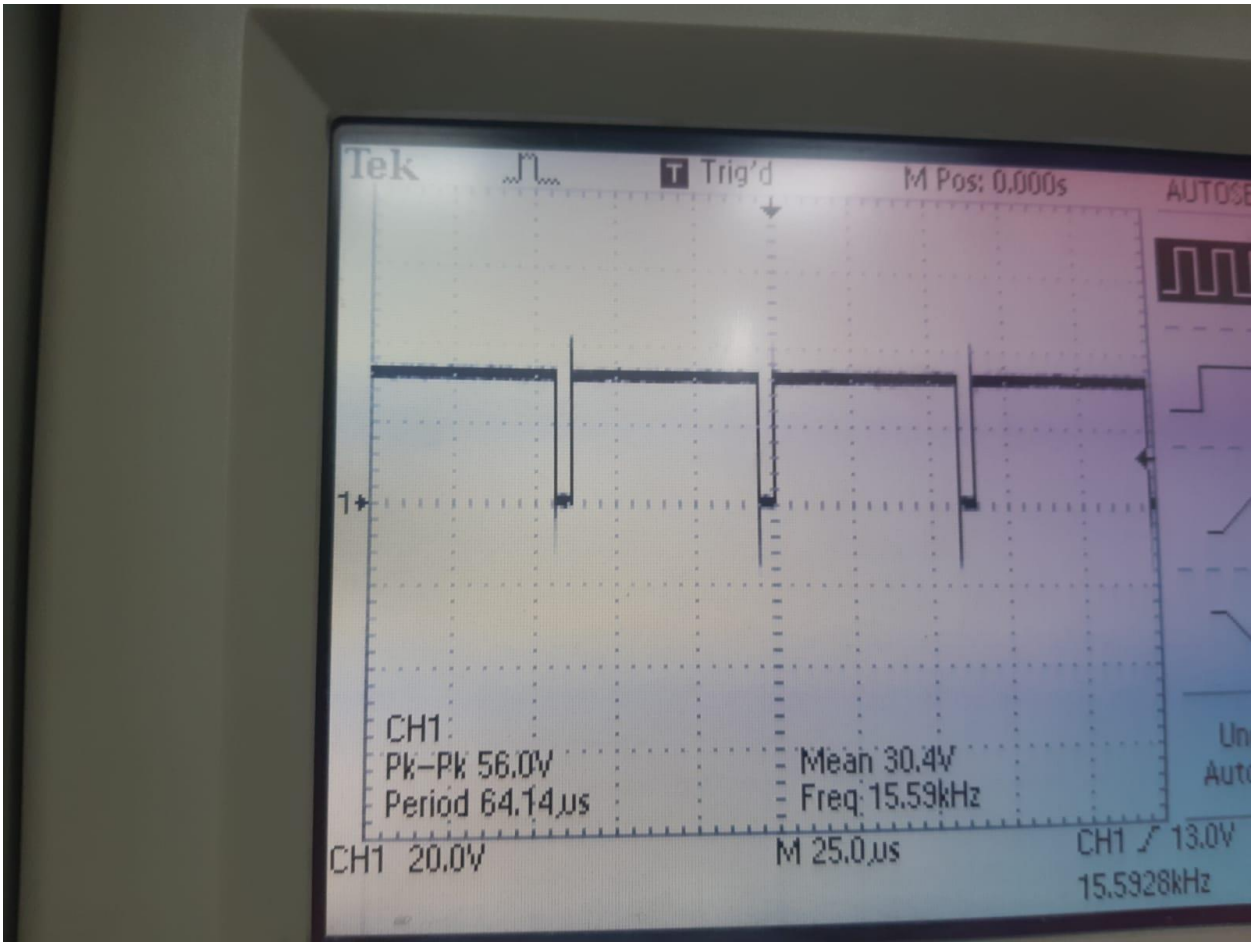
All the interrupts in use are for the LCD, we didn’t use interrupts for joystick and timer0

צילום של הסקופ:

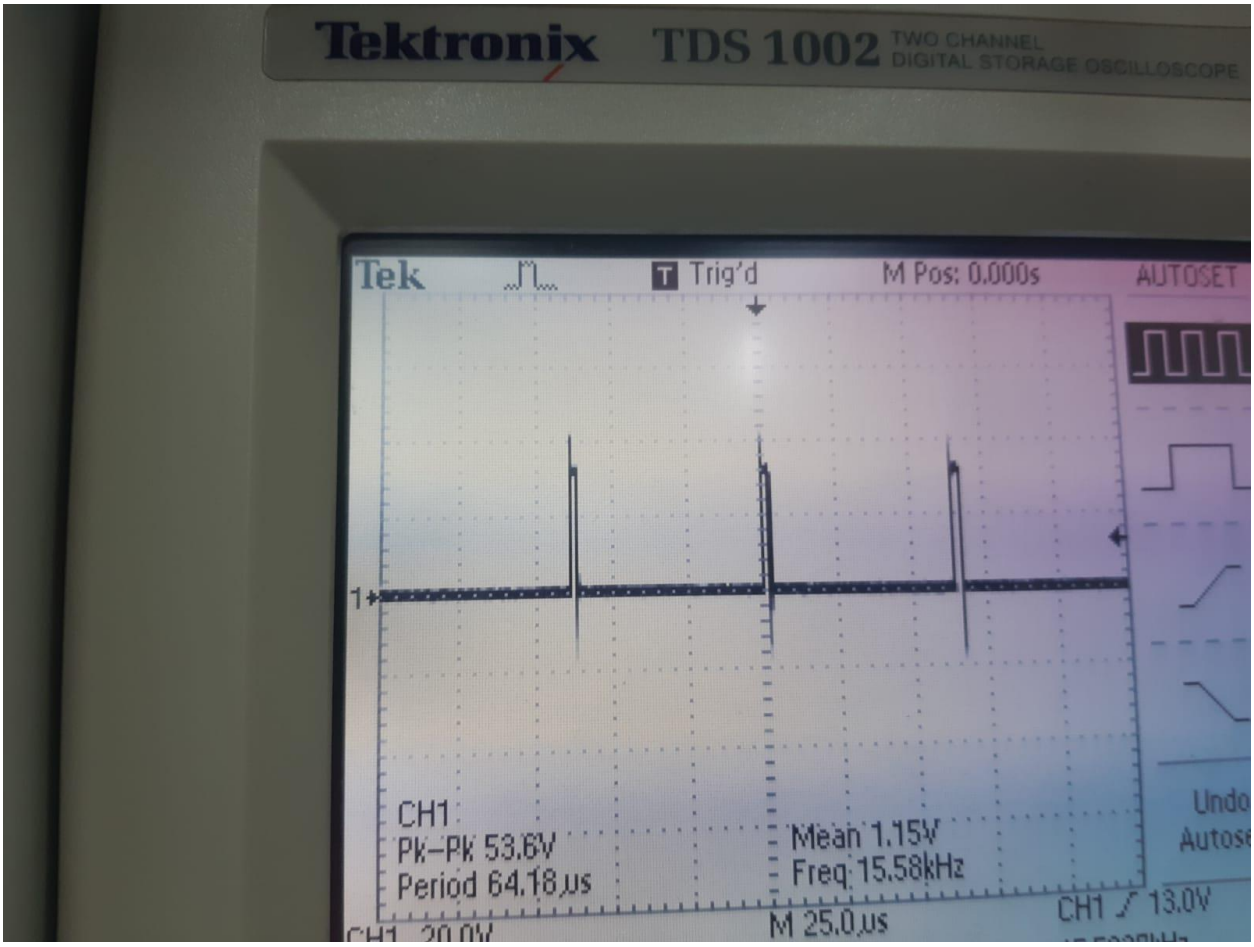
PWM 50%



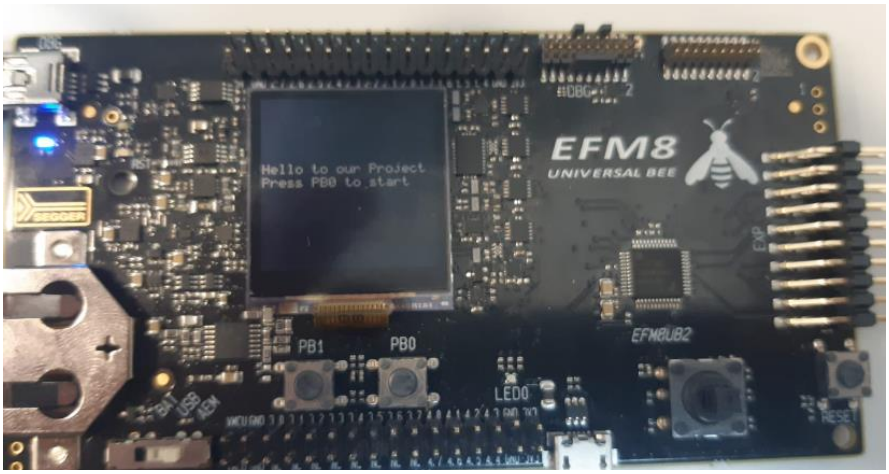
PWM almost 100% (when move joystick to east)



PWM almost 0% ((when move joystick to west)



צילום של תוצאות הבקר:



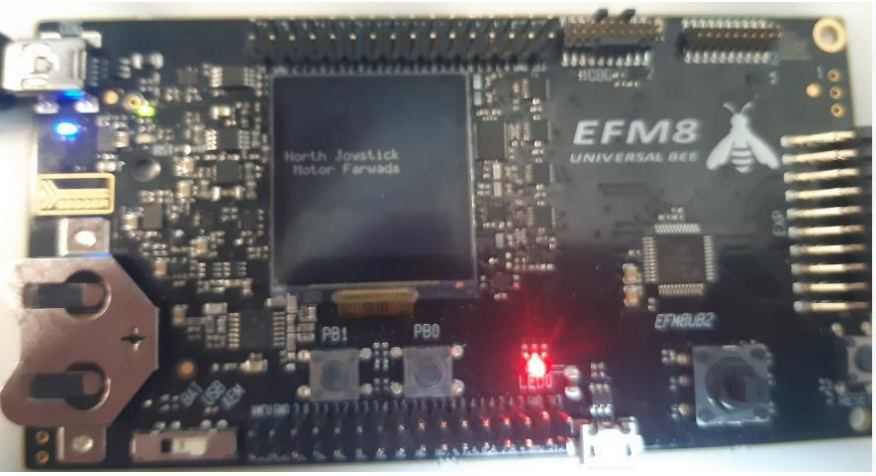
We start the program, we see that the Leds are off, and get LCD message.
The motor is off

Press LCD : Hello to our project PB0 to start

We Press PB0, we see that the Green led is on, and get LCD message.

The motor is off

LCD: Start Move Joystick

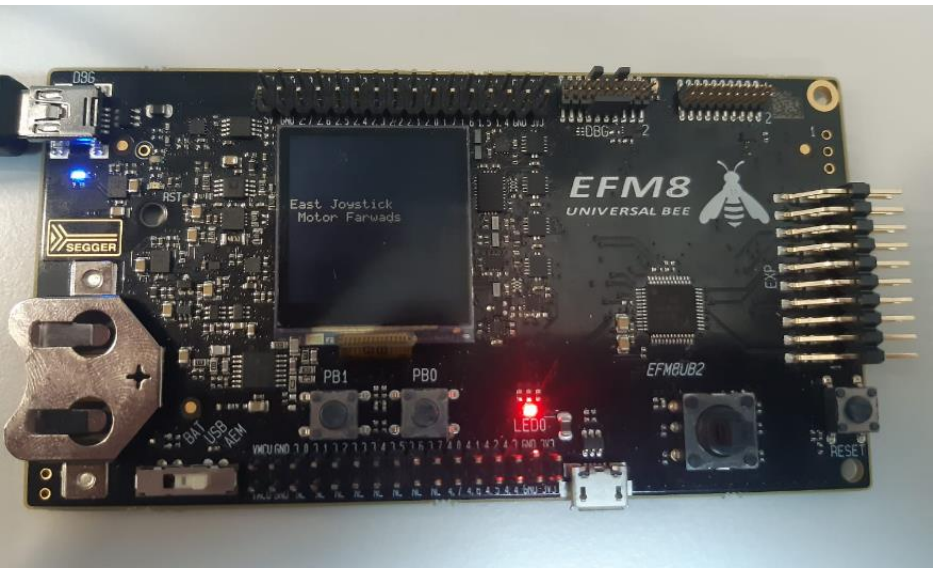


We Move joystick north, we see that the RED led is on, and get LCD message.

Motor moves farwards

LCD: North Joystick

motor Farwards



We Move joystick east, we see that the RED led is on, and get LCD message.

Motor moves farwards and increase speed

LCD: East Joystick

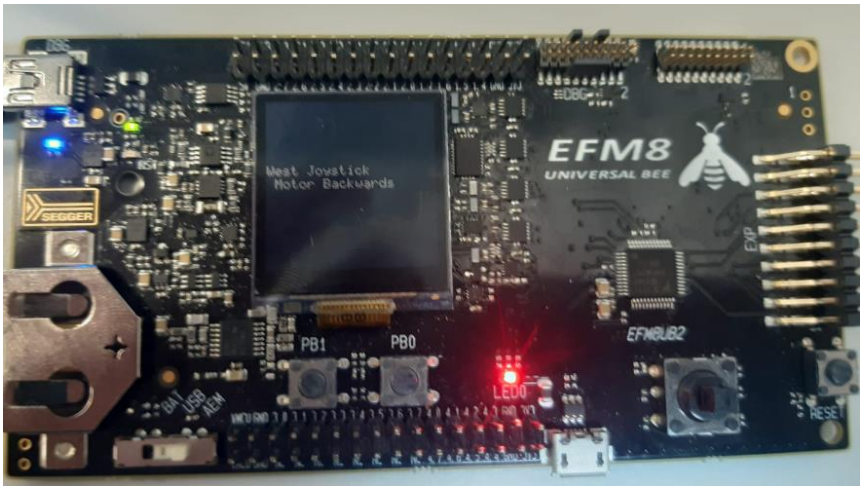
motor Farwards

We Move joystick south, we see that the RED led is on, and get LCD message.

Motor moves backwards

LCD: South Joystick

motor Backwards

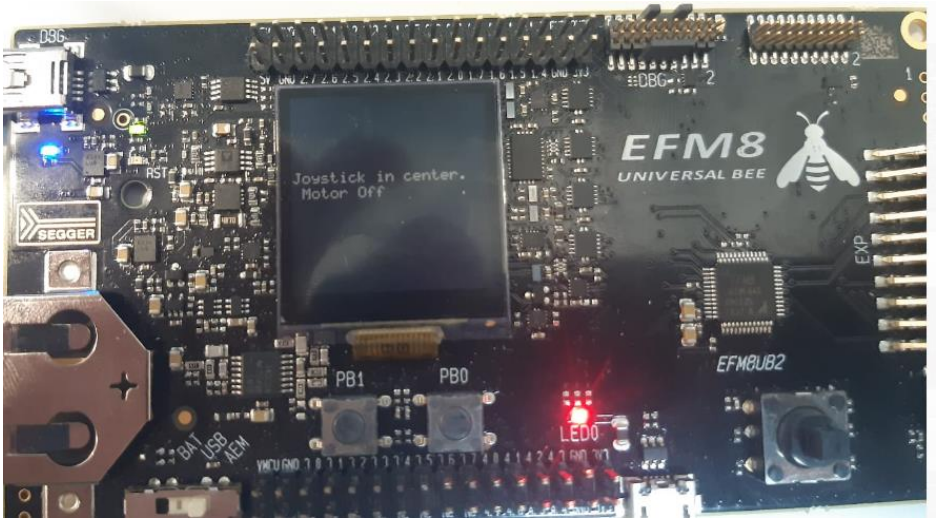


We Move joystick west, we see that the RED led is on, and get LCD message.

Motor moves farwards and decrease the speed

LCD: West Joystick

motor Backwards



We Move joystick to center, we see that the RED led blinking, and get LCD message.

Motor is off

LCD: Joystick in center
Motor off

We press PB1, then the motor is off, and the Leds are off and we got LCD message

LCD: PB1 is pressed

Motor off

תיאור תקלות שהתעוררו במהלך התכנון:

לקח לנו הרבה זמן עד שגילנו שלא יכולים לשנות את קובץ הפסיקות של הLCD, כי היינו מתכננים את העבודה והקוד שלנו היה כבר כתוב עם פסיקות, היה רק צריך להוסיף את כתיבה למסך. אז עד שידענו שאי אפשר להוסיף פסיקות לקובץ LCD, אז תכננו את זה עוד פעם בלי פסיקות באותו קובץ של ה LCD.