Light Tracker

Juan Camilo Salgado Ramirez
Faculty of Electronic Engineering
Universidad de Ibagué
Ibagué, Colombia
2420181027@estudiantesunibague.edu.co

Steve Daniel Rincon Sandoval
Faculty of Electronic Engineering
Universidad de Ibagué
Ibagué, Colombia
2420181043@estudiantesunibague.edu.co

***Abstract*— The project will be carried out in MPLAB X on a PIC16F877A for a light tracker car. This system will be used in a car but it can be used in static objects such as solar panels since its objective is to position / follow the largest source of light.**

Keywords—PIC16F15244, light tracker, language C++, MPLAB X

I. Introduction

This project presents a light tracker implemented in a car, but it can be implemented in other objects that need to be oriented to the greatest light source such as certain silver, solar panels. It seeks to optimize the operation of objects that need light as well as to reduce the manual load of calibrating, orienting or moving the aforementioned objects so that they are in their optimal state.

II. Methodology

A. Flow Chart

In order to carry out the project, a flow chart must be created to facilitate the order in the programming and the logic in it.

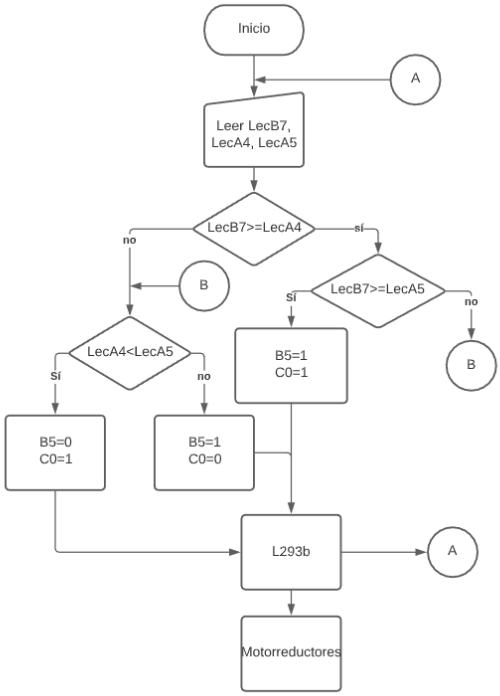

Fig. 1. Flow Chart

B. Logic

With the previous diagram you begin to program everything you need to put the flow diagram into a code and thus obtained results.

III. Results

Once programmed the following fragmented code was obtained.

A. PIN Manager Initialize

In this part of the code, the ports were started as inputs or outputs as needed, as well as defining whether these would be digital or analog.

```
// TRISx registers
TRISB5 = 0;
TRISC0 = 0;

// ANSELx registers
ANSELBbits.ANSB7 = 1;
ANSELAbits.ANSA4 = 1;
ANSELAbits.ANSA5 = 1;
```

Fig. 2. PIN MANAGER Initialize

B. Read all inputs

All entries are read one by one. in this way, when an input is read, its value is saved individually so that it is not mixed or interchanged with another and then it is unified in a single register..

```
void Read_ADC(void){
    ADCON0= 0b00010001;
    ADCON0bits.GO = 1;
    while (ADCON0bits.GO);
    resultHigh = ADRESH;
    resultLow = ADRESL;
    Read();
    LecA4 = Lectura;
    // Switch to reading A5
    ADCON0= 0b00010101;
    ADCON0bits.GO = 1;
    while (ADCON0bits.GO);
    resultHigh = ADRESH;
    resultLow = ADRESL;
    Read();
    LecA5 = Lectura;
    // Switch to reading B7
    ADCON0= 0b00111101;
    ADCON0bits.GO = 1;
    while (ADCON0bits.GO);
    resultHigh = ADRESH;
    resultLow = ADRESL;
    Read();
    LecB7 = Lectura;
}
```

Fig. 3. Read ADC

C. Calculate

The values read at the inputs are compared to be able to send instructions to the l293b driver as chosen. In the first case the sensor located in front receives the greatest amount of light therefore it advances, in the second case a lateral sensor receives the lightest therefore rotates towards the direction of

the sensor and finally the sensor on the opposite side receives the lightest therefore it rotates towards the direction of the side. In this way it will always be located in front of the greatest amount of light.

```
void Calculate(){
    if((LecB7>=LecA4)&&(LecB7>=LecA5)){
        LATBbits.LATB5 = 1;
        LATCbits.LATC0 = 1;
    }
    else if(LecA4<LecA5)
    {   LATBbits.LATB5 = 0;
        LATCbits.LATC0 = 1;
    }
    else{
        LATBbits.LATB5 = 1;
        LATCbits.LATC0 = 0;
    }
}
```

Fig. 4. Calculate

D. Main

The PIN Manager is initialized as well as the oscillator. A "while" is created so that in this way the inputs are constantly being read while the project is on, as well as constantly making comparisons between the sensors to know what order to send to the gearmotor driver.

```
void main(void){
    PIN_MANAGER_Initialize();
    OSCILLATOR_Initialize();
    ADC();
    while(1){
        Read_ADC();
        Calculate();
        __delay_ms(60);
    }
}
```

Fig. 5. Main

IV. Conclusions

The following conclusions were obtained regarding the realization of the project.

A.

B.

C.

References

[1] PIC16F15244 Curiosity Nano Hardware User Guide. 2020 [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/PIC16F15244-Curiosity-Nano-Hardware-User-Guide-DS50003045A.pdf. [Accessed: 26- Nov- 2020]

[2] PIC16F15213/14/23/24/43/44 Full-Featured 8/14/20-Pin Microcontrollers. 2020 [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/PIC16F15213-14-23-24-43-44-Data-Sheet-DS40002195B.pdf. [Accessed: 26- Nov- 2020]