FUNDAMENTAL OF AUTOMATION

"SOLAR TRACKER"

SUBMITTED BY:

Sparsh Goud (2020BTechCSE077)

Vikas Janu(2020BTechCSE083)

SUBMITTED TO:

Dr. Devika Kataria

Dr. Hanuman Prasad Agrawal

Institute of Engineering and Technology (IET)

JK Lakshmipat University Jaipur

15 JULY, 2021

CERTIFICATE

This is to certify that the project work entitled "**Solar Tracker**" submitted by **SPARSH GOUD(2020BTechCSE077) ,Vikas Janu(2020BTechCSE083)** towards the partial fulfillment of the requirements for the degree of **Bachelor of Technology in Engineering** of JK Lakshmipat University Jaipur is the record of work carried out by them under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted.

Dr. Hanuman Prasad Agarwal
Assistant Professor
Department of Electrical & Electronics Engineering
Institute of Engineering & Technology (IET)
JK Lakshmipat University Jaipur

Dr. Devika Kataria
Associate Professor
Department of Electrical & Electronics Engineering
Institute of Engineering & Technology (IET)
JK Lakshmipat University Jaipur

Date of Submission: 15 JULY 2021

ACKNOWLEDGEMENT

We wish to express our sincere gratitude to Vice Chancellor DR. Dheeraj Sanghi sir, Dean Dr. Sanjay Goel sir, and special thanks to Dr. Hanuman Prasad Agarwal Sir and Dr. Devika Kataria ma'am for their guidance and encouragement in carrying out this project work, and we would like to thank some of our friends for helping us in some difficult situation while working on this project.

Sincerely yours

Sparsh Goud(2020BTechCSE077)

Vikas Janu(2020BTechCSE083)

TABLE OF CONTENTS

Problem statement/ Ideation:

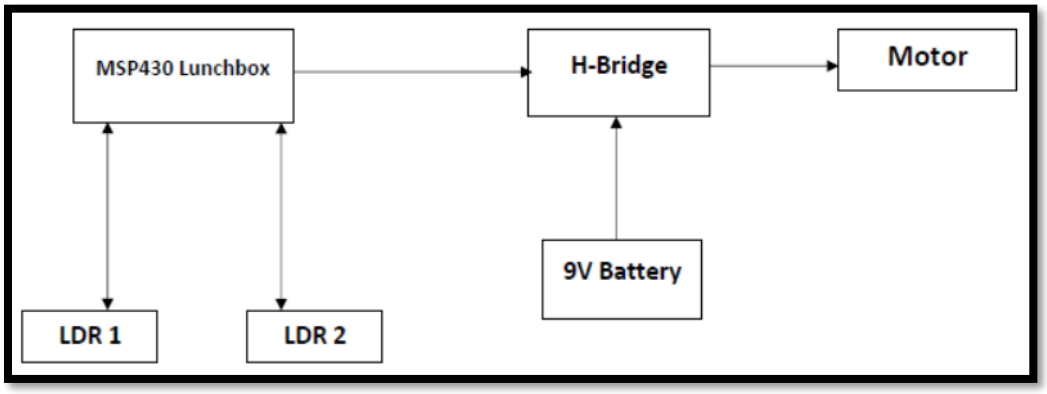
A solar tracker device has a wide scope of uses to improve harnessing of sun based disconnection. The issue presented hence is to execute a framework that is fit for further developing solar power production . A microcontroller is utilized to carry out the control circuit which thus positions an engine used to orient the sun powered board ideally .

SIMULATOR USED:

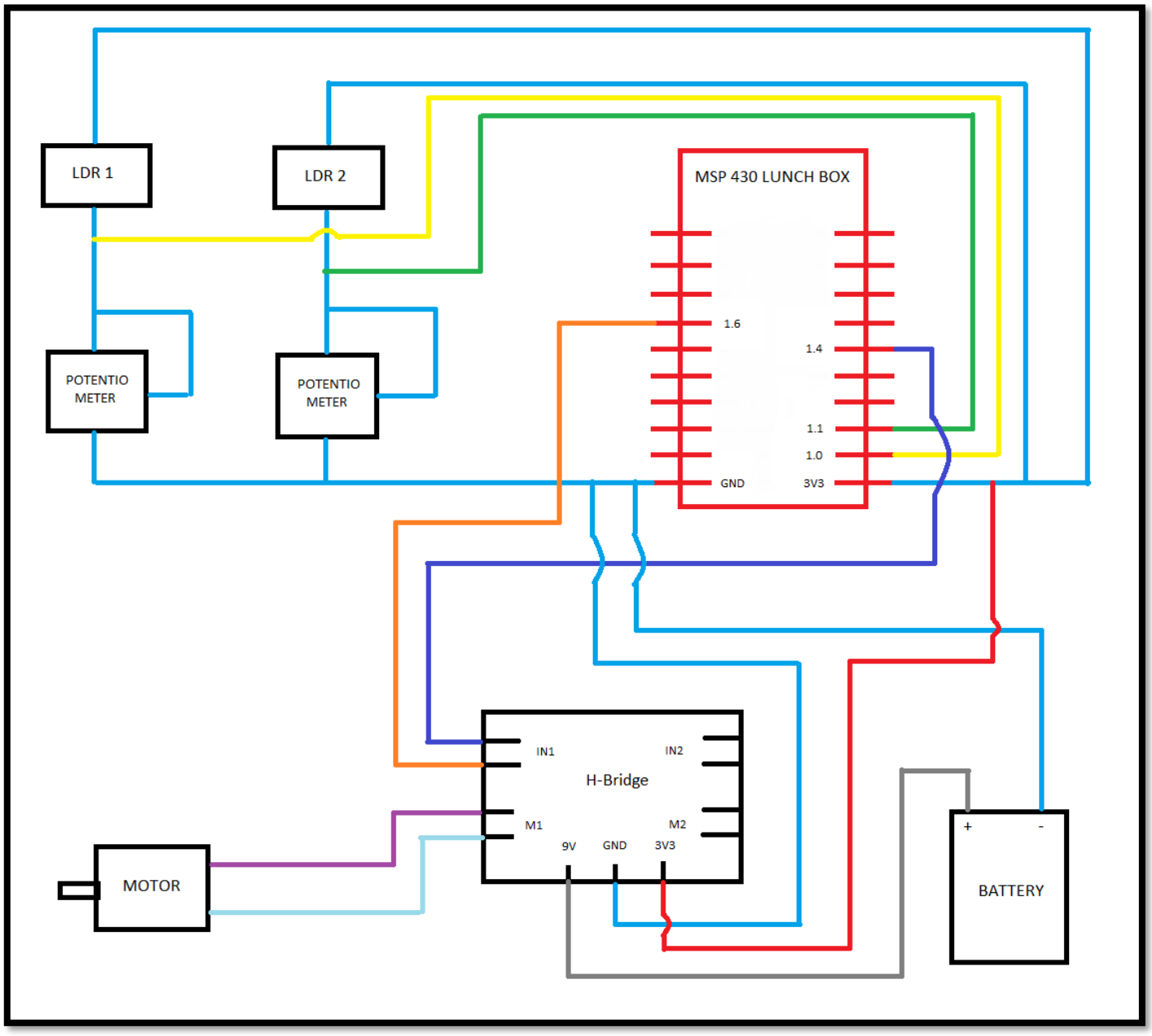
Code Composer Studio: Code Composer Studio is an Integrated Development Environment based on Eclipse which is used for working with each and every Texas Instruments (TI) digital offering - MSP430 MCUs, Tiva MCUs, C2000 MCUs, Hercules MCUs, AM335x processors, C55x DSPs (and all other DSP families), the DMx processors etc.

The software includes an optimizing C/C++ compiler, source code editor, project build environment, debugger, profiler and many other features. The intuitive IDE provides a single-user interface that takes you through each step of the application development flow. Familiar tools and interfaces let you get started faster than ever before. Code Composer Studio software combines the advantages of the Eclipse software framework with advanced embedded-debug capabilities from TI resulting in a compelling feature-rich development environment for embedded developers.

Methodology/Flowchart:



Schematic Diagram:



Program Code:

```
#include "msp430.h"

#define ADC_CHANNELS 2

unsigned int samples[ADC_CHANNELS];

#define LED1 BIT4

#define LED2 BIT6

#define SENSOR_LEFT BIT0

#define SENSOR_GND BIT2

#define SENSOR_RIGHT BIT1

#define SENSOR_GND1 BIT3

#define RED_LED LED1

#define GRN_LED LED2

void ConfigureAdc(void){

ADC10CTL1 = INCH_1 | ADC10DIV_0 | CONSEQ_3 | SHS_0;

ADC10CTL0 = SREF_0 | ADC10SHT_2 | MSC | ADC10ON | ADC10IE;

ADC10AE0 =SENSOR_LEFT + SENSOR_RIGHT ;

ADC10DTC1 = ADC_CHANNELS;

}

void main(void) {

WDTCTL = WDTPW | WDTHOLD;

BCSCTL1 = CALBC1_1MHZ;

DCOCTL = CALDCO_1MHZ;

BCSCTL2 &= ~(DIVS_3);

P1DIR = 0; /* set as inputs */

P1SEL = 0; /* set as digital I/Os */

P1OUT = 0; /* set resistors as pull-downs */

P1REN = 0xFF; /* enable pull-down resistors */
```

```
P2DIR = 0; /* set as inputs */
P2SEL = 0; /* set as digital I/Os */
P2OUT = 0; /* set resistors as pull-downs */
P2REN = 0xFF; /* enable pull-down resistors */
P1REN &= ~(LED1 | LED2); /* disable pull-up/downs */
P1DIR |= (LED1 | LED2); /* configure as oututs */
P1REN &= ~(SENSOR_GND |SENSOR_GND1); /* disable pull-up/down */
P1OUT &= ~(SENSOR_GND|SENSOR_GND); /* SENSOR_GND should be at GND */
P1DIR |= (SENSOR_GND |SENSOR_GND1); /* SENSOR_GND must be an output */
P1REN |= (SENSOR_LEFT|SENSOR_RIGHT); /* enable pull-up on SENSOR */
P1IN |= (SENSOR_LEFT|SENSOR_RIGHT); /* set resistor as pull-up */
ConfigureAdc();
__enable_interrupt();
while (1) {
__delay_cycles(1000);
ADC10CTL0 &= ~ENC;
while (ADC10CTL1 & BUSY);
ADC10SA = (unsigned int)samples;
ADC10CTL0 |= ENC + ADC10SC;
__bis_SR_register(CPUOFF + GIE);
if (samples[0] < samples[1]) {
P1OUT |=RED_LED;
P1OUT &= ~(GRN_LED);
} else if (samples[0] == samples[1]) {
P1OUT &= ~(RED_LED);
P1OUT &= ~(GRN_LED);
} else {
```

```
P1OUT |= GRN_LED;

P1OUT &= ~(RED_LED);

}

}

}

#pragma vector=ADC10_VECTOR

__interrupt void ADC10_ISR (void){

__bic_SR_register_on_exit(CPUOFF);

}
```

Results:

Both LDR are connected with potentiometer which helps in keeping a constant potential drop across both LDR so that when our msp430 is connected it does not any sudden movement so when LDR1 receives more light than LDR2 , it offers lower resistance than LDR1 providing a high input to L293D H-bridge motor driver . As a result output pin of L293D H-bridge motor driver goes high to rotate motor in one direction says (clockwise) and turn the solar panel.

when LDR2 receives more light than LDR1 , it offers lower resistance than LDR2 providing a high input to L293D H-bridge motor driver . As a result output pin of L293D H-bridge motor driver goes high to rotate motor in one direction says (anti-clockwise) and turn the solar panel.