

**Activity 4.2.8b Satellite Flight Data ROBOTC Program**

Introduction

Use the text below to recreate your own program. This program will be used to instruct the robot to acquire elevation data.

Equipment

* Computer with ROBOTC software

Procedure

1. Open the ROBOTC software.
2. Click File then New File.
3. The program body should be empty. If it is not then select all the text in the main program window and press the delete key on the keyboard.
4. Select the text below and copy into the blank ROBOTC program.

#pragma config(Sensor, dgtl1, trackend, sensorTouch)

#pragma config(Sensor, dgtl2, sonar, sensorSONAR\_inch)

#pragma config(Sensor, dgtl4, quad, sensorQuadEncoder)

#pragma config(Motor, port2, drive, tmotorNormal, openLoop)

//\*!!Code automatically generated by 'ROBOTC' configuration wizard !!\*//

/\*

 Project Title: Activity 4.2.8b Satellite Flight ROBOTC Program

 Team Members: AE Curriculum Team

 Date: TBA

 Section: TBA

 Task Description:

 Operates a robot to gather elevation data.

 Pseudocode:

 Wait 2 seconds before program task start to allow time to be clear of the satellite after powering on the Cortex

 Wait until joystick 6U button pushed.

 Start a motor to traverse bar.

 while the touch sensor isn't pressed signifying that the satellite has not reach the end of the frame ...

 Wait for the target number of encoder counts to elapse and monitor that satellite has not reach the end of the frame.

 (2.75 in. dia, therefore 360 counts / 3.14 (2.75 in.) = 41.69 counts/in.)

 (3 in. travel = 125 counts)

 Output wheel encoder and sonar data data in streaming window for export into excel

 stop the motor

\*/

task main()

{

 wait1Msec(2000); // Robot waits for 2000 milliseconds before executing program

 SensorValue[quad]=0; // Set optical shaft encoder quadrature count to zero

 int targetCount = 1;

 while(vexRT[Btn6U] == 0) //Satellite waits until the button is pressed before moving on

 {

 }

 startMotor(drive, -64); //Turn on the motor in reverse at 64

 while(SensorValue[trackend] == 0) //While the final button isn't pressed...

 {

 while(abs(SensorValue[quad]) < targetCount && SensorValue[trackend] == 0)//Wait for the encoder count to reach the target polling point (125 counts or 3 inches of travel) and satellite has not reached end of the path

 {

 }

 writeDebugStreamLine("%d \t %d", SensorValue(quad), SensorValue(sonar)); //...and then write optical shaft encoder and sonar values to the debug stream

 targetCount = targetCount + 125; // Sensor data is written every 125 counts or 3 inches of travel

 }

 stopMotor(drive); //Stop the motor

} // End task main