

Creating an Obstacle Avoidance Tank with an MSP430, IR Sensor, and Motor Driver

Darlan Schwarz

Harmandeep Kaur

Florida Atlantic University

Author's Note

This project was conducted as a final exam, for Professor Ravi Shankar's Microprocessor course, in the College of Engineering and Computer Science, at Florida Atlantic University.

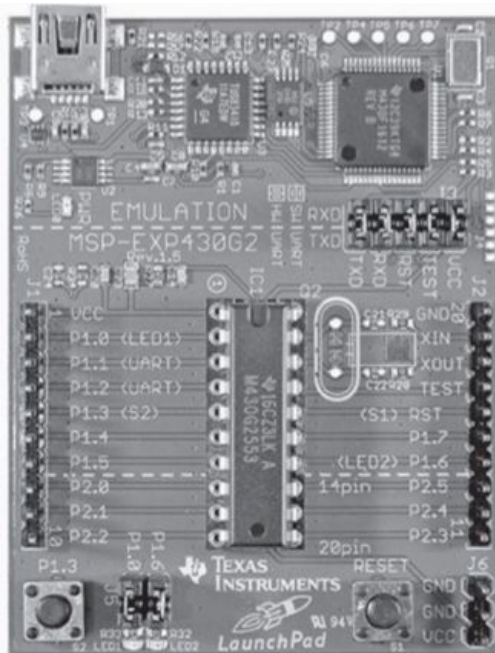
Abstract

The goal of this project is to be able to utilize the TI microcontroller, MSP430g2553 in a co-design of software and hardware for real life problems. Unsalan Cem offers the idea for the Obstacle Avoidance Tank, in “Programmable Microcontrollers with Applications”. So we have set out to create a motorized vehicle that will detect and avoid obstacles in its path. Using the MSP430, an infrared sensor, and two controlled DC motors and C programming we implement the application. Bar a few complications, our vehicle is able to successfully navigate obstacles in its path. This application has not reached its full potential yet; it can be further enhanced with the addition of features for efficiency, such as, a rear sensor and an all wheel drive.

Key words: MSP430, Obstacle Avoidance Tank, Infrared sensor, DC motor, C programming.

Background

The MSP430g2553 is a TI, 16 bit, RISC, low power, and inexpensive microcontroller. With 16 general-purpose pins that range from digital input and output to digital communications, it makes for a great tool to have for a variety of different uses. For this project we will focus on the analog to digital conversions, ADC, and pulse width modulation modules, PWM. The ADC allows for the interfacing of the infrared sensor with the microcontroller, by gathering physical data and digitizing it, so one can then use it to program a customized reaction for any given situation. The PWM is used then to synthesize a digital to analog conversion, by allowing one to regulate how a physical response is delivered through the microcontroller.

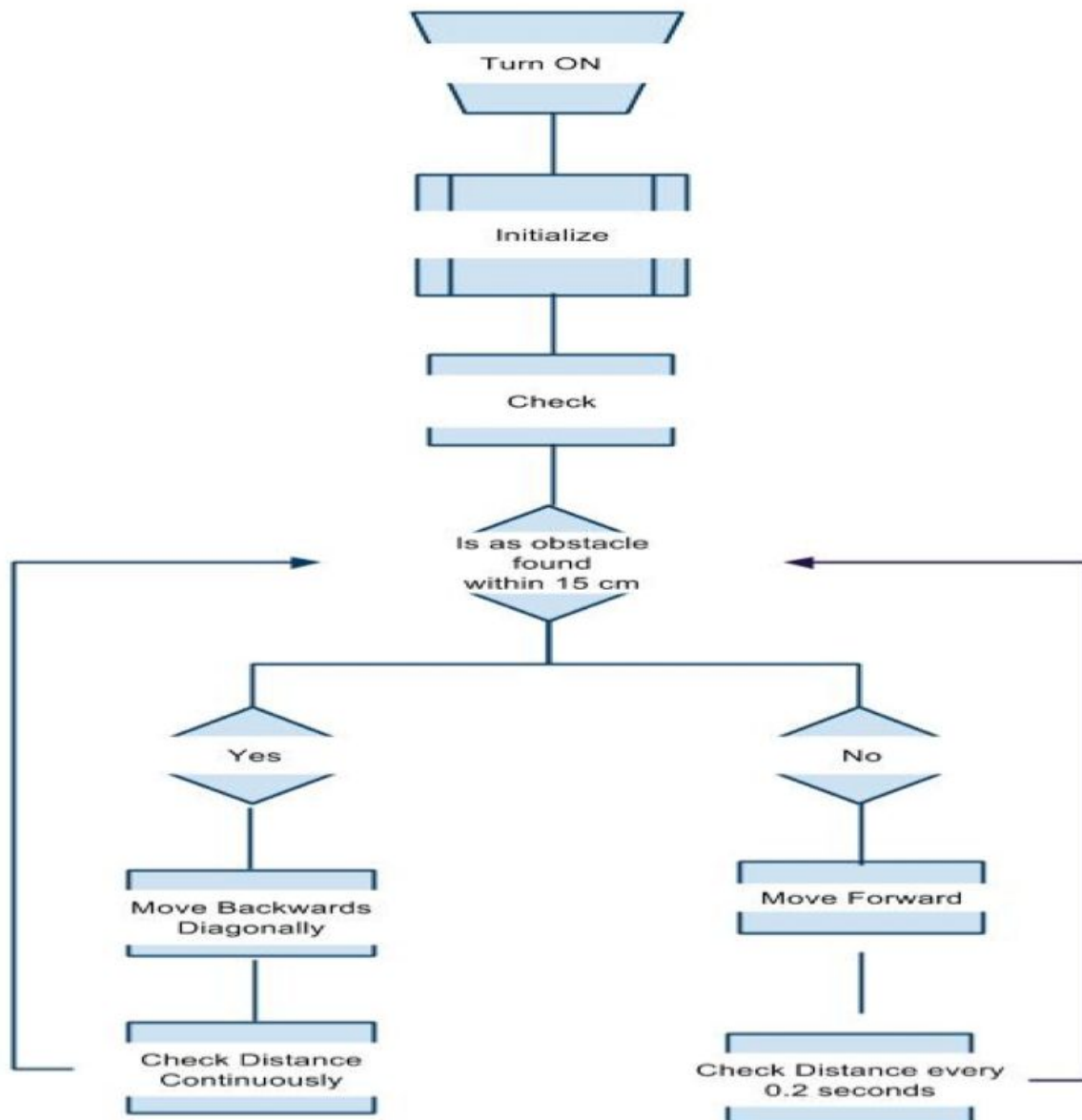


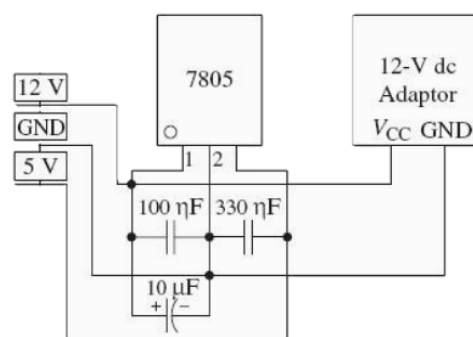
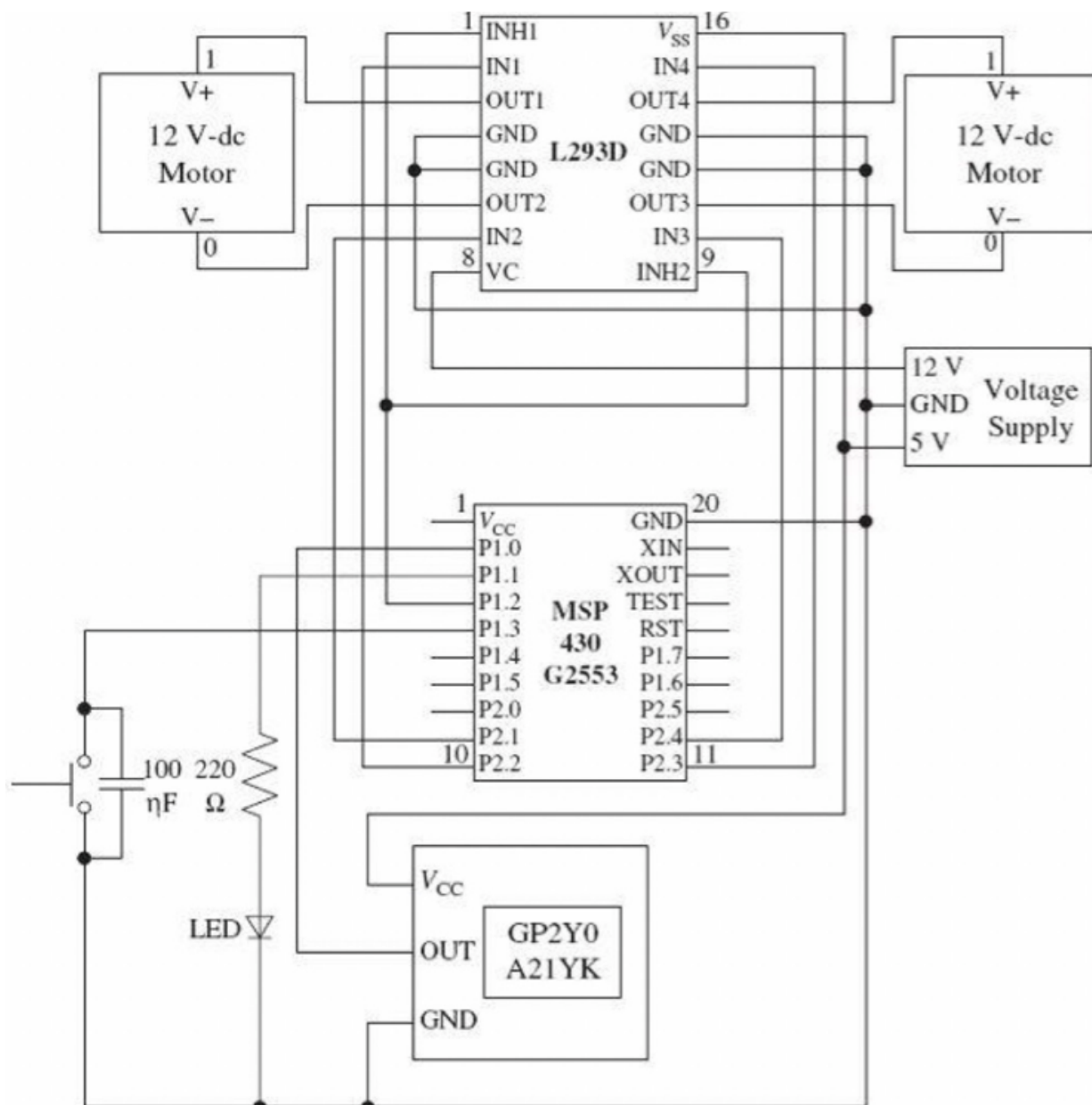
Pin	Port Name	Usage Area
1	V _{CC}	Source voltage
2	P1.0	Digital I/O, Interrupt, Timer, ADC
3	P1.1	Digital I/O, Interrupt, Timer, ADC, Digital Communication
4	P1.2	Digital I/O, Interrupt, Timer, ADC, Digital Communication
5	P1.3	Digital I/O, Interrupt, ADC
6	P1.4	Digital I/O, Interrupt, Timer, ADC, Digital Communication
7	P1.5	Digital I/O, Interrupt, Timer, ADC, Digital Communication
8	P2.0	Digital I/O, Interrupt, Timer
9	P2.1	Digital I/O, Interrupt, Timer
10	P2.2	Digital I/O, Interrupt, Timer
11	P2.3	Digital I/O, Interrupt, Timer
12	P2.4	Digital I/O, Interrupt, Timer
13	P2.5	Digital I/O, Interrupt, Timer
14	P1.6	Digital I/O, Interrupt, Timer, ADC, Digital Communication
15	P1.7	Digital I/O, Interrupt, ADC, Digital Communication
16	Reset	Non-maskable interrupt
17		
18	P2.7	Digital I/O, Interrupt, External Crystal
19	P2.6	Digital I/O, Interrupt, Timer, External Crystal
20	V _{SS}	Ground voltage

Methods

The obstacle avoidance tank will be designed with a proximity sensor and two dc motors. The tank will check for an obstacle every 0.2 s, if no obstacle is found within 15 cm the tank will move. Rotating the dc motors in the same direction with a suitable PWM signal will have the tank move forward at a relatively fast pace. Yet if an obstacle is ever detected by the front facing sensor within 15 cm of the vehicle, it will

reverse diagonally. One of the dc motors will be stopped while the other is motor is moving counter clockwise with a slower PWM signal. All while continuously checking the obstacle's distance until it is no longer within the 15 cm threshold. Therefore the tank will continue to move forward now in that direction and return to checking for obstacles every 0.2 s. The tank also features a switch to power the entire tank on and off.





The two major components in the schematic, excluding the MSP430, are the motor driver, L293D, and the IR sensor. The L293D is a quadruple half h driver allowing for up to two bi-directional motors. We used an alternative IR sensor, the GP2Y0A41SK0F, rather than the GP2Y0A21YK. The sensor has a light emitter and detector with a triangulation method and outputs analog data via voltage to indicate proximity, the closer the object the higher the voltage outputted.

Results

This project found that the individual functions worked accurately as separate pieces but when used in conjunction with each other many discrepancies occurred. We believe this is due to spikes in voltage around the board fluctuating. The current range is from 520 to 600 and to reduce the change in fluxuation we have bumped this up to 530 at the risk of reducing the range. When including the PWM for the motors it successfully regulates the speed for the motors but the signal creates a disturbance that allows the motors to turn on without prompting. Lastly the weight of the tank is too much for what the wheels are currently outputting or the motors are unable to exert enough force to move the vehicle. So in order to test our theory to turn was done by creating a lift for the tank on a plate that allowed the tank to slide in the direction it would turn towards.

CCS Debug - Obstacle Avoiding Tank/main.c - Code Composer Studio

File Edit View Project Tools Run Scripts Window Help

Quick Access CCS Edit CCS Debug

Debug

Obstacle Avoiding Tank [Code Composer Studio - Device Debugging]

TI MSP430 USB1/MSP430 (Suspended - SW Breakpoint)

- getanalogvalues() at main.c:109 0xC236
- main() at main.c:38 0xC02E
- _c_int00_noargs_noexit() at boot_special.c:103 0xC2F2 (the entry point was reached)

Expression

Expression	Type	Value	Address
distance	int	588	0x020E
ADCReading	int[3]	0x0216	0x0216
speed	int	50	0x0208
Add new expression			

main.c IR sensor working

```
103 ADC10CTL0 |= (ENC | ADC10SC);           //Start ADC Conversion
104 while (ADC10CTL1 & BUSY);               //Wait while ADC is busy
105 distance += ADCReading[1];
106 // power += ADCReading[2];
107 }
108 distance = distance/5;                   // Average the 5 reading for the three variables
109
110
111
```

Console

Obstacle Avoiding Tank

Writable Smart Insert 109:1 Free License

CCS Debug - Obstacle Avoiding Tank/main.c - Code Composer Studio

File Edit View Project Tools Run Scripts Window Help

Quick Access CCS Edit CCS Debug

Debug

Obstacle Avoiding Tank [Code Composer Studio - Device Debugging]

TI MSP430 USB1/MSP430 (Suspended - SW Breakpoint)

- getanalogvalues() at main.c:109 0xC236
- main() at main.c:38 0xC02E
- _c_int00_noargs_noexit() at boot_special.c:103 0xC2F2 (the entry point was reached)

Expression

Expression	Type	Value	Address
distance	int	512	0x020E
ADCReading	int[3]	0x0216	0x0216
speed	int	50	0x0208
Add new expression			

main.c IR sensor working

```
103 ADC10CTL0 |= (ENC | ADC10SC);           //Start ADC Conversion
104 while (ADC10CTL1 & BUSY);               //Wait while ADC is busy
105 distance += ADCReading[1];
106 // power += ADCReading[2];
107 }
108 distance = distance/5;                   // Average the 5 reading for the three variables
109
110
111
```

Console

Obstacle Avoiding Tank

Writable Smart Insert 109:1 Free License

Discussion

We initially followed the schematic exactly as it was but ran into a few issues. Therefore it was decided to make some minor changes with the pins and opted for switches rather than push buttons. There was a challenging learning curve in regards to the mixed signals, which required more time than anticipated or allotted. Pressed for time with many rising complications resulted in an incomplete project with none of the added functionality we hoped for. Some basic improvements would be a rear sensor, all wheel drive, connected axels, and more intuitive speed variations. At its current functionality the tank has motor control to go forward, reverse, and maneuver its direction to avoid obstacles in front of it. It also has two speed functions a fast for when it detects no object and slow for when it does find an obstacle in its path.

As is the state of the tank now it may be used in a strategy game where an opponent would seek to trap or slow down the tank. But with increased functionality it can be used as a safe pathfinder, delivery robot, vehicle-to-vehicle communication and autonomous vehicles.

Conclusion

The obstacle avoidance tank can successfully navigate objects directly in its path and has great potential in how it can be further applied as a solution to real life problems. From solving traffic issues to detecting toxin levels in environments lethal for life, this vehicle can navigate itself safely to solve these problems. The use of multiple signals allows us to customize solutions and for many real life applications.

References

Unsalan, Cem. *Programmable Microcontrollers with Applications: MSP430 LaunchPad with CCS and Grace*. McGraw-Hill Education, 28 Aug. 2013. Web. 12 Dec. 2016.

SHARP. "GP2Y0A41SK0F." *SHARP GP2Y0A41SK0F IR Ranger Sensor (4-30cm) SKU:SEN0143 - Robot Wiki*. Web. 6 Dec. 2016.

"SHARP GP2Y0A41SK0F IR Ranger Sensor (4-30cm) SKU:SEN0143." *SHARP GP2Y0A41SK0F IR Ranger Sensor (4-30cm) SKU:SEN0143 - Robot Wiki*. 26 June 2015. Web. 12 Dec. 2016.

"L293x Quadruple Half-H Drivers (Rev. D)." *SHARP GP2Y0A41SK0F IR Ranger Sensor (4-30cm) SKU:SEN0143 - Robot Wiki*. Texas Instruments Incorporated, Jan. 2016. Web. 8 Dec. 2016.