

#ThugLifeRobot





Hey there folks,

Today i am going to show you guys how to use RC controllers and decode their signals for any of your projects

As many of the you know that we can directly use servo motors with RC receivers since, servos use PPM(Pulse Position Modulation) signals as inputs to operate ,so by decoding the signals we can use RC receivers and transmitters to create a variety of wireless projects ..

So to demonstrate this we shall use a 4 wheel drive (4WD) robot .. to drive the wheels we are using four DC motors ,and they are controlled using a motor driver which operates on PWM(Pulse Width Modulation) signals. We will discuss the difference between PWM and PPM in upcoming steps.

So lets get started !!!!!

 Add Tip  Ask Question  Comment  Download



Teacher Notes

Teachers! Did you use this instructable in your classroom?
Add a Teacher Note to share how you incorporated it into your lesson.

Add Teacher Note

RF Technologies

OPEN

advertisement

Step 1: Before Getting Started





3 More Images

Following are the materials required :

- Arduino uno (you can google it,there are plenty of places you can get it from)
- RC Transmitter (i am using flysky fst6 transmitter and receiver you can get it at various RC hobby websites for about 50\$)
- RC receiver(i am using flysky r6b)
- Motor driver (i am using MD30 dual channel motor driver,i got this driver form 2krobotics.com for testing purposes ,they actually works great !!!)
- male to female jumpers
- 4 DC motors
- 1 Robot chassis (although not necessary -you can make one :))
- 12 volt battery (we are using 3 cell li-po battery)

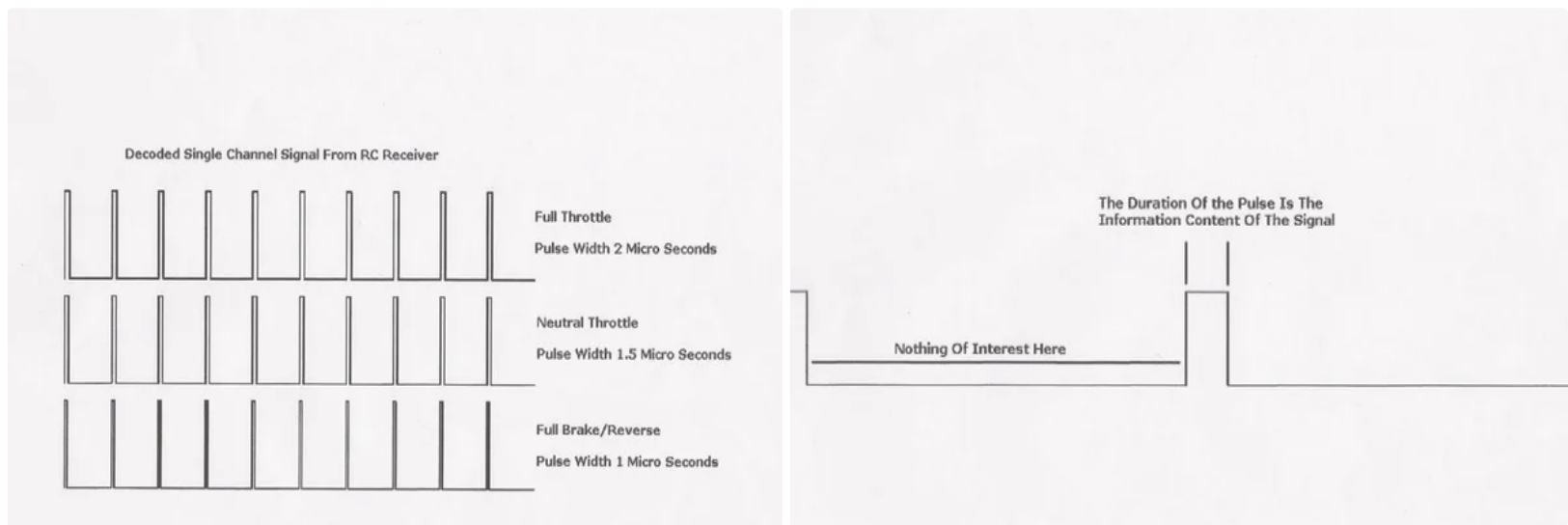
tools required

- screw driver (flat head and Philips screw driver)

Prerequisite

This tutorial is assuming that you are familiar with arduino development platform

Step 2: Difference Between Ppm and Pwm



PWM and PPM are two common words used in the R/C industry. PWM stands for Pulse Width Modulation and PPM stands for Pulse Position Modulation. Some devices that use PWM for control are ESC's (electronic speed controls) and servos. PWM is a technique used to relay data in the form of a varying pulse width.

You may be already familiar with binary, 1's and 0's; where a 1 is represented as 'on' and a 0 as 'off'. An example of this would be a light switch. Turning the switch on would indicate a 1, off a 0. In the case of a PWM/PPM signal, a voltage applied indicates a 1 and vice versa. However, in the case of R/C electronics this 'on/off' data is not enough, this is where the pulse width comes in.

The way we relay data to a servo for instance is the time the pulse is on. In the case of R/C electronics this time is usually around 1-2 milliseconds. A servo or ESC will monitor this pulse and begin counting when the pulse is detected and stop counting when the pulse stops. The time the pulse is on will determine the servo position. For example, sending a servo a 1ms pulse will make the servo swing completely left while a 2ms pulse will swing the arm completely right.

Generally in R/C equipment an entire PWM pulse will last a total of 20ms. The entire pulse is called a frame. A complete frame will include both the time the pulse is high (1-2ms) and the time the pulse is low. The image below represents a typical PWM frame.

Aside from the gaining servo holding power, the reason for the 20ms frame is just having the ability to line up several PWM signals in the same frame. Like I said before, the time the pulse is on is what is important because we are able to strip out this relevant data from a PPM frame to re-generate a PWM frame. For example, if a radio only sent 1 PWM signal at a time, it would take 20ms per channel. If you have an 8 channel radio each update would take 160ms. The same data can be packed into a PPM frame and only take 20ms per update. Transmitters and receivers are the two most common R/C devices that use PPM.

The following are lists of common devices that use PPM and PWM.

R/C Devices that use PWM Pulses:

Servos

Electronic Speed Controllers

R/C switches

R/C lights

R/C receivers

Data loggers

Failsafe's

Autopilot/Stabilization systems

Servo Controller

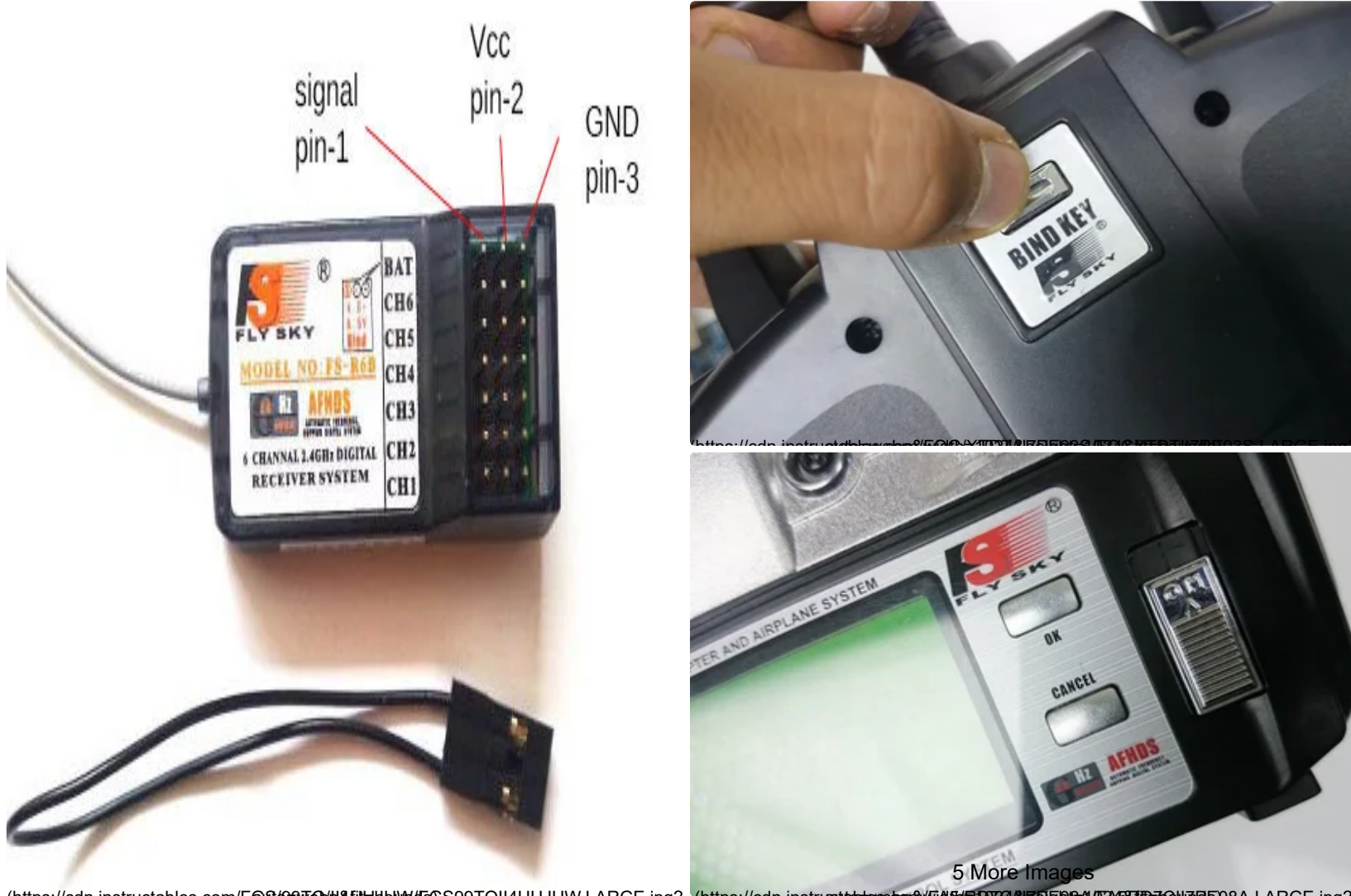
R/C Devices that use PPM Pulses:

R/C transmitters

R/C receiversAutopilot/Stabilization systems

PCTx

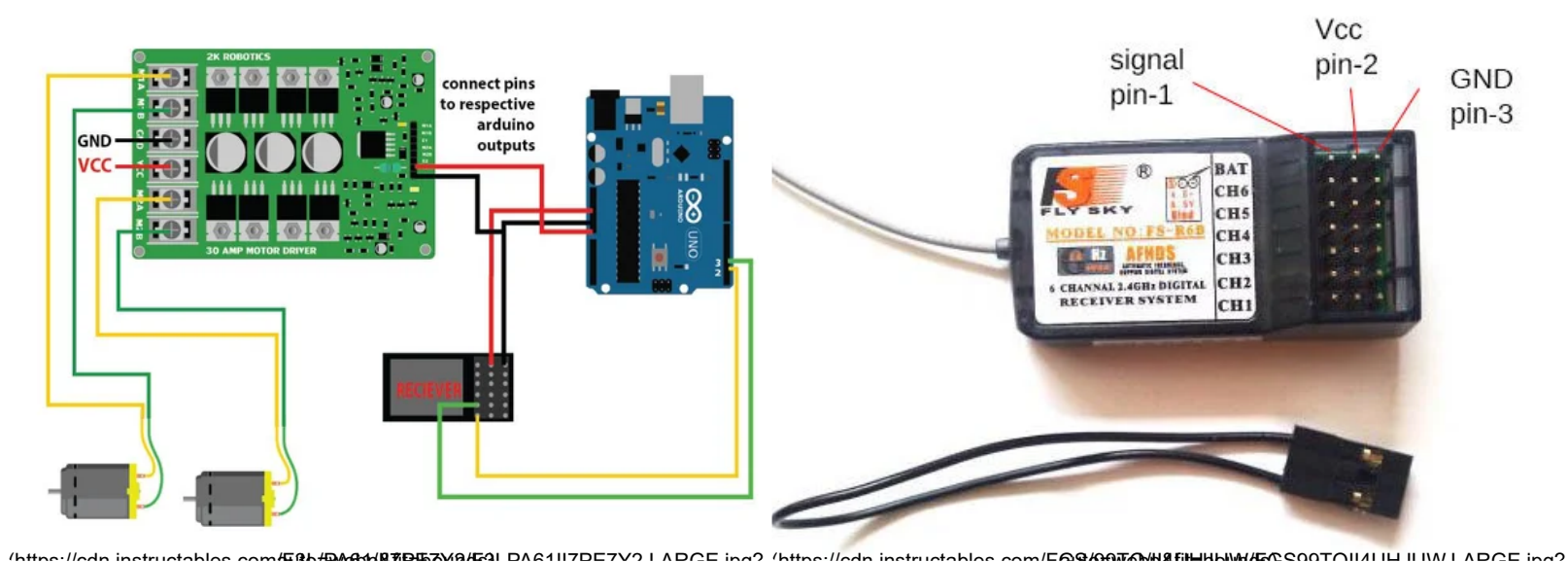
Step 3: Binding Receiver(RX) and Transmitter(TX)



following are the steps to bind RX and TX :

1. Insert binding plug in the receiver's BAT port (if you do not have one you can use female to female jumper and connect pin 1 and 3 of RX together)
2. Connect pin 2 and 3 any row of the receiver to +5V and GND of arduino (providing 5v supply to receiver)
3. Power up the arduino !
4. You will notice that receiver's light starts to blinking ,this shows that your receiver is not bound to any transmitter.
5. Next, at the back of transmitter you will find the bind key,press and hold the bind key and switch the transmitter ON. **Note :transmitter should be OFF for all the above steps and should turn ON ONLY after you press bind key**
6. Now if your receiver's LED status is a solid light then you have successfully bound the transmitter to your receiver.
7. If all the above steps goes right then you are done lets head on !!!

Step 4: Wire It Up and Calibrate !



connect all the modules and components together (refer to image above for connections also listed below)

connections are :

MOTOR DRIVER -> ADUINO

- M1A -> arduino pin5
- M1B ->arduino pin6
- ENABLE1 ->arduino pin10
- M2A ->arduino pin7
- M2B ->arduino pin8
- ENABLE2 ->arduino pin11
- GND-> arduino gnd
- +5v ->arduino vcc

RECEIVER -> ARDUINO

- CH1 -> pin 3
- CH2 -> pin 2
- GND->GND
- +5V->+5V

attach motors to respective slots on motor driver

also power everything up by connecting battery to GND and VCC of motor driver

Burning the code

- download the attached file and open it in arduino ide
- select the board type (tools->board->Arduino uno)
- select the COM port (tools->port-> *to the port your arduino is connected to*)
- click on compile (to check any code errors ,though it is error free but it is good to check always ;))
- burn the code by clicking on upload button

Calibration time !!

- generally the PPM signal gives a value of 1000 to 2000 as we saw in PPM and PWM step
- but that is ideal value of an ideal remote(TX)
- so we need to calibrate it

follow the steps below to calibrate your remote

1. after uploading the code to arduino and all wiring done ,click on serial monitor in arduino IDE
2. note idle value and pen down the values of idle
3. move throttle up and down and also pen down those values also
4. in code to the top you can see " #define THROTTLE_SIGNAL_IN_PIN 2 " in line 2
5. there pin 2 refers to pin connected arduino pin 2
6. to check the calibrate the staring just change "2" to "3" after calibration for throttle is done
7. repeat step 1,2 &3 for staring also
8. now calibration part is done !!!



Add Tip



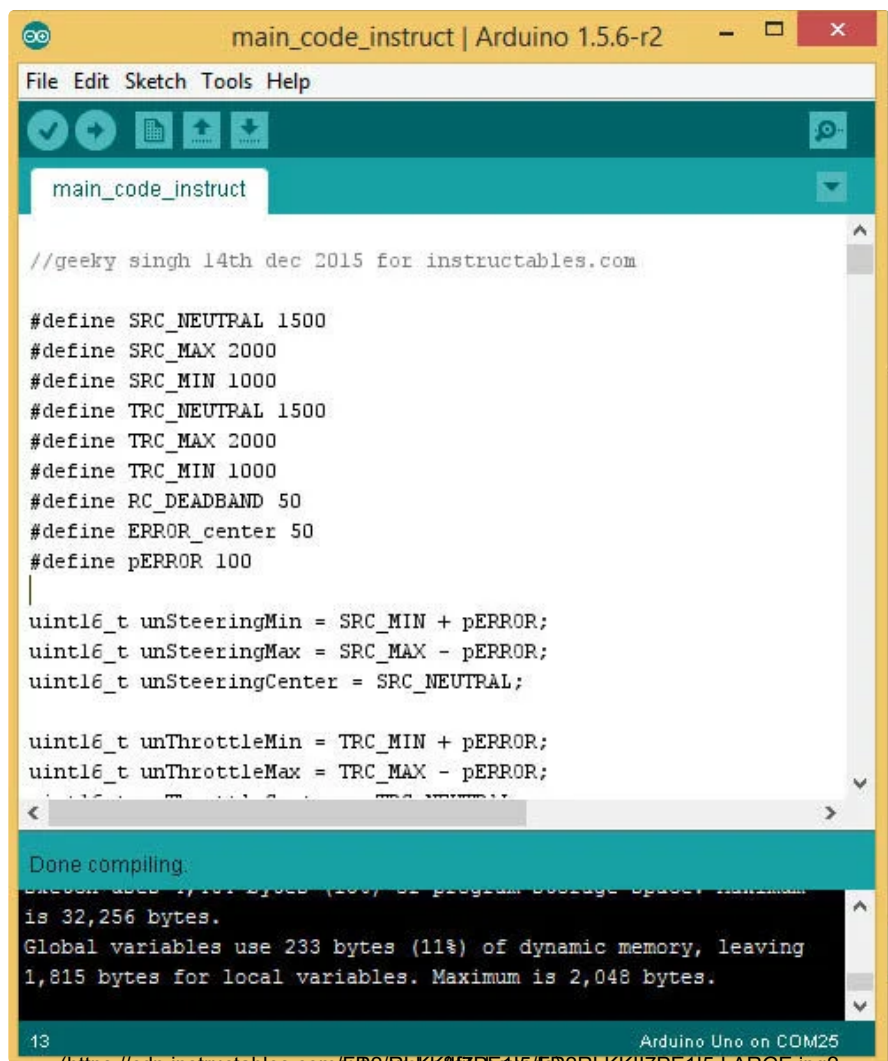
Ask Question



Comment

Download

Step 5: Burning the Main Code



well then since we are done with the calibration part and we have our actual transmitter MAX , MIN and IDLE values for staring and throttle we are good to upload the actual code to arduino we just need to replace these values to the values we got while calibration(i have use ideal values for referance purpose)

//stearing max min and neutral

#define SRC_NEUTRAL 1500
#define SRC_MAX 2000

#define SRC_MIN 1000

//throttle max min and neutral

#define TRC_NEUTRAL 1500
#define TRC_MAX 2000
#define TRC_MIN 1000

main_code_instruct.ino

Upload (https://cdn.instructables.com/FF7/61AL/II7PDOCE/FF761ALII7PDOCE.ino)
Download (https://cdn.instructables.com/FF7/61AL/II7PDOCE/FF761ALII7PDOCE.ino)

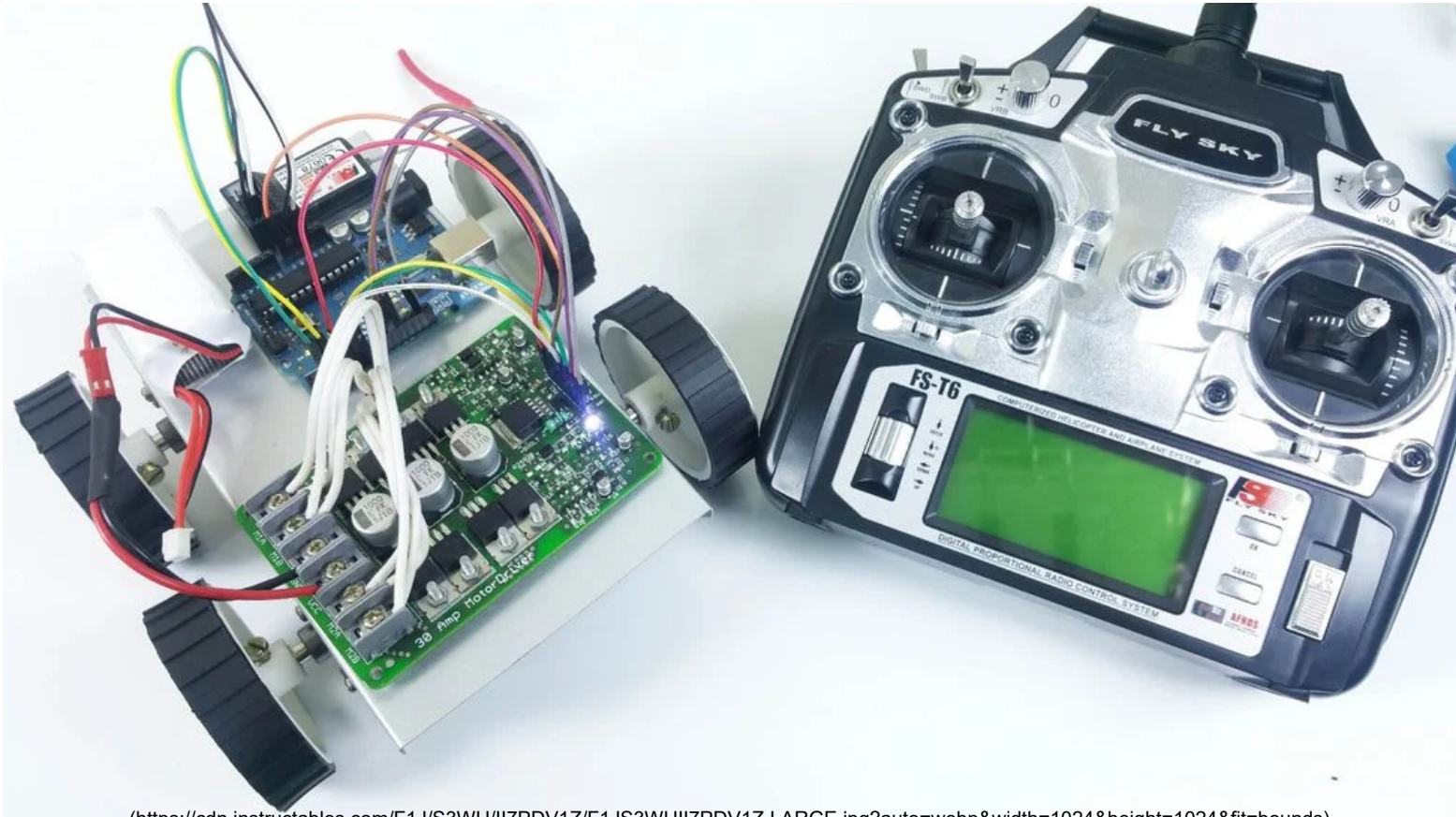
Add Tip

Ask Question

Comment

Download

Step 6: Last Step






(https://cdn.instructables.com/F51/JS2W/L/1117BDV/17/F51JS2W/L/1117BDV/17/LABCE.jpg?auto=webp&width=1024&height=1024&fit=bound)


well the final step is ... to make something you like to build with this hack !!

hope you like it

comment below if you like it ,and also if you want any more elaboration on any topic related to this project ,feel free to ask

thank you for reading this instructable !!!

 Add Tip  Ask Question  Comment Download



CONTEST

TECH

HobbyKing

Participated in the
[Tech Contest \(/contest/HobbyKing/\)](/contest/HobbyKing/)

View Contest



CONTEST

ARDUINO

ALL THE THINGS