POBOTS Team 353

Implementing Mecanum Drive in LabVIEW

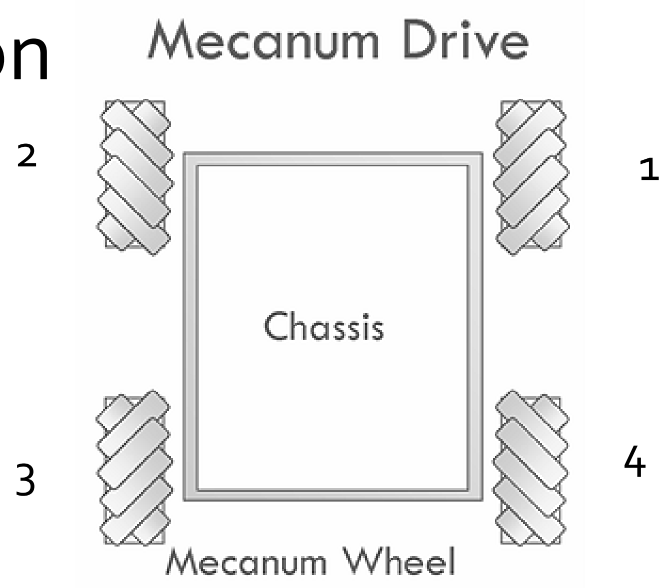
pobots.com

Overview

- LabVIEW contains libraries that make implementing mecanum very simple
- We are going to discuss a method of programming with the following controls:
    - Left joystick for translational motion
    - Left shoulder buttons for CCW rotation
    - Right shoulder buttons for CW rotation

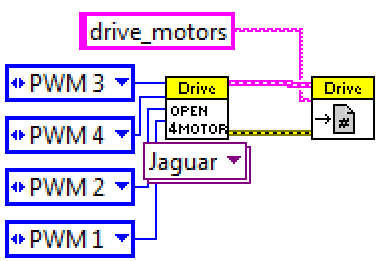
Step 1: Checking the wheel orientation

- Check that the wheels appear in the "X" orientation from a top view
- Note the number of the wheels--they will be used as convention in the rest of the presentation

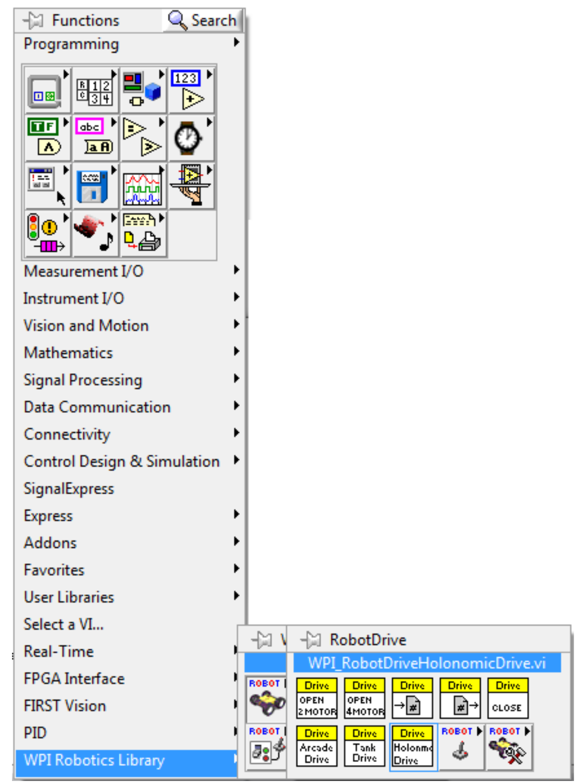


Step 2: Wiring the basic Holonomic VI

- Create a new FRC Robot Project
- In Begin.vi, open four motors
- This tutorial will assume that PWM1 is top right, PWM2 is top left, PWM3 is rear left, and PWM4 is rear right

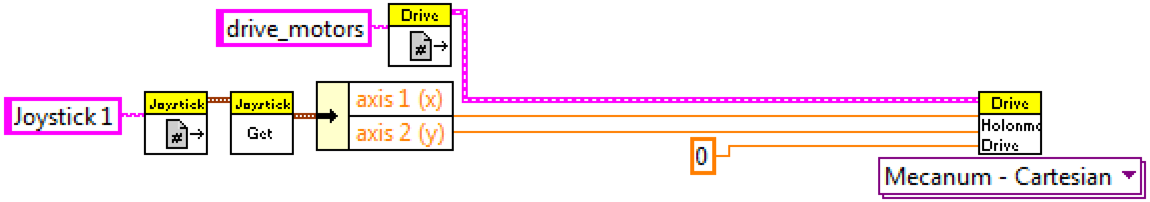


- In Teleop.vi, find the Holonomic VI



Step 2: Wiring the basic Holonomic VI

- Place this code in Teleop.vi



- We are setting rotation to 0 to begin to test the basic wheel motions-feel free to wire up rotation to a third joystick axis (such as the throttle, or a second controller)

Step 3: Benchtesting

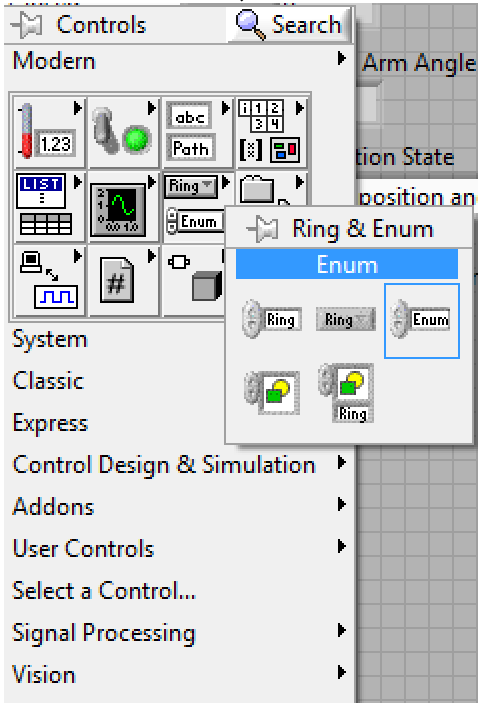
- Place your robot on blocks, so that the wheels don't touch the floor
- Run the program, and perform the following tests

Joystick Direction	Desired output
Joystick North	Wheels 1-4 forward
Joystick South	Wheels 1-4 backwards
Joystick East	Wheels 2,4 forward; 1,3 backwards
Joystick West	Wheels 1,3 forward; 2,4 backwards

- If problems exist, invert motors or switch PWM channels as necessary

Step 4: Implementing Rotation

- Let's add some code for rotation
- Open Robot Global Data.vi, and create an enumerated variable (name it Rotation Mode)



Step 4: Implementing Rotation

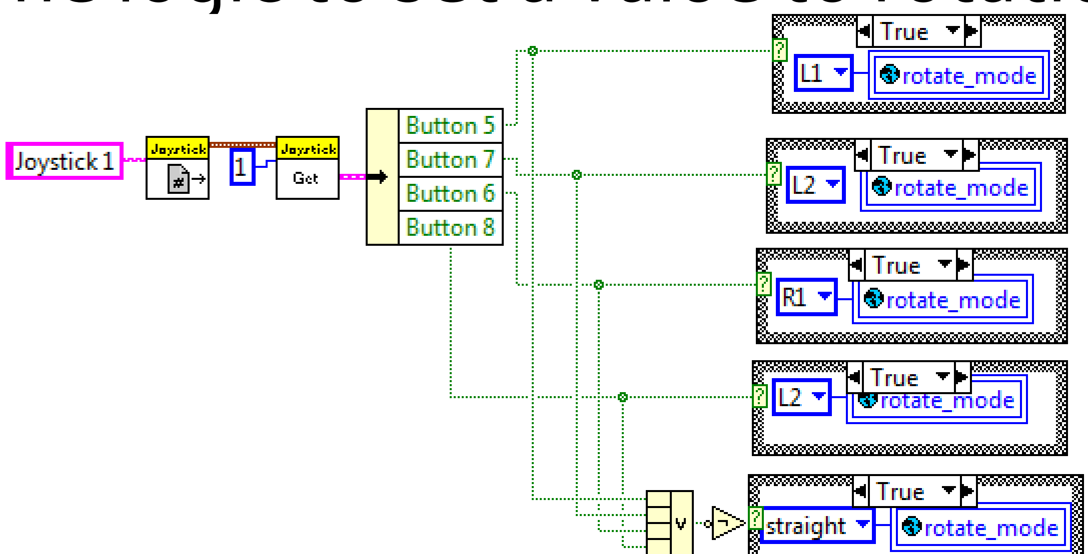
- Right click on the variable, and go to "Edit Items"
- Add the following items:

Items	Digital Display
straight	0
L1	1
L2	2
R1	3
R2	4

- R1 and L1 will control slow rotation; R2 and L2 will control fast rotation

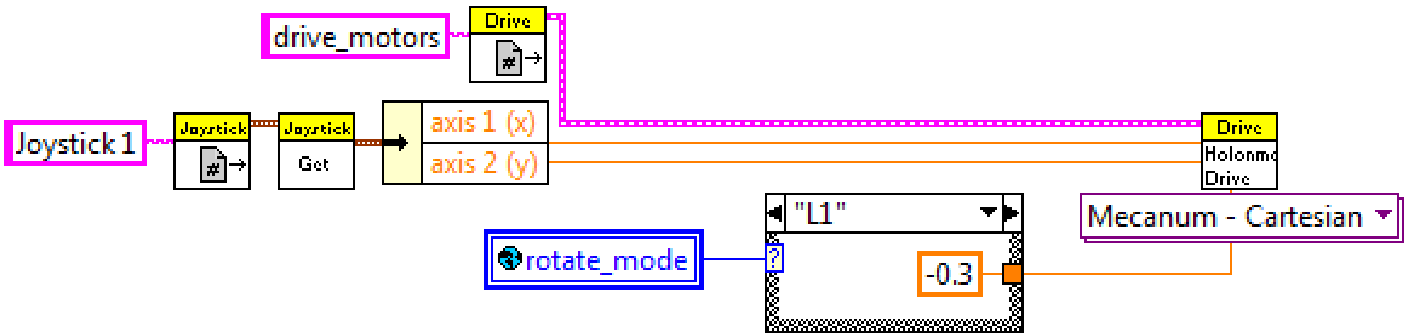
Step 4: Implementing Rotation

- Find out the numbers corresponding to each shoulder button in Control Panel (for Windows)
  - We will assume 5=L1, 6=R1, 7=L2, and 8=R2
- Write some logic to set a value to rotation mode



Step 4: Implementing Rotation

- Now, let's act upon the rotation mode



- As shown, use -0.3 for L1
- Use -0.6 for L2, 0.3 for R1, 0.6 for R2, and 0 for straight

Step 5: Floor Test

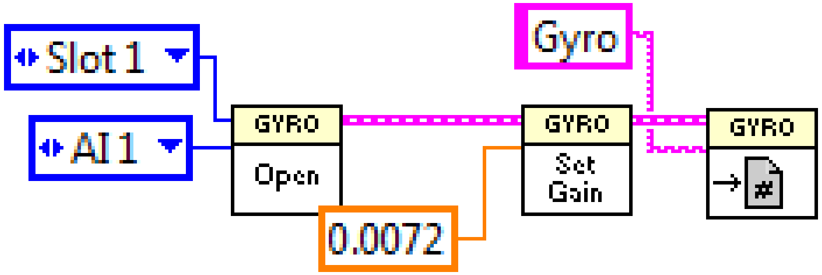
- Now it's time to place the robot on the floor and see how the robot drives
- The robot should move--albeit poorly--in all translational and rotational directions
- When you try to translate without rotation, however, you should notice that the robot will rotate anyway-let's fix this

Step 6: Maintaining a fixed header

- We will accomplish this with **closed loop control**
  - This means that we will "close the loop" by comparing the desired output with the actual output and correct for differences
- To measure the actual output, we need to use a gyro

Step 6: Maintaining a fixed header

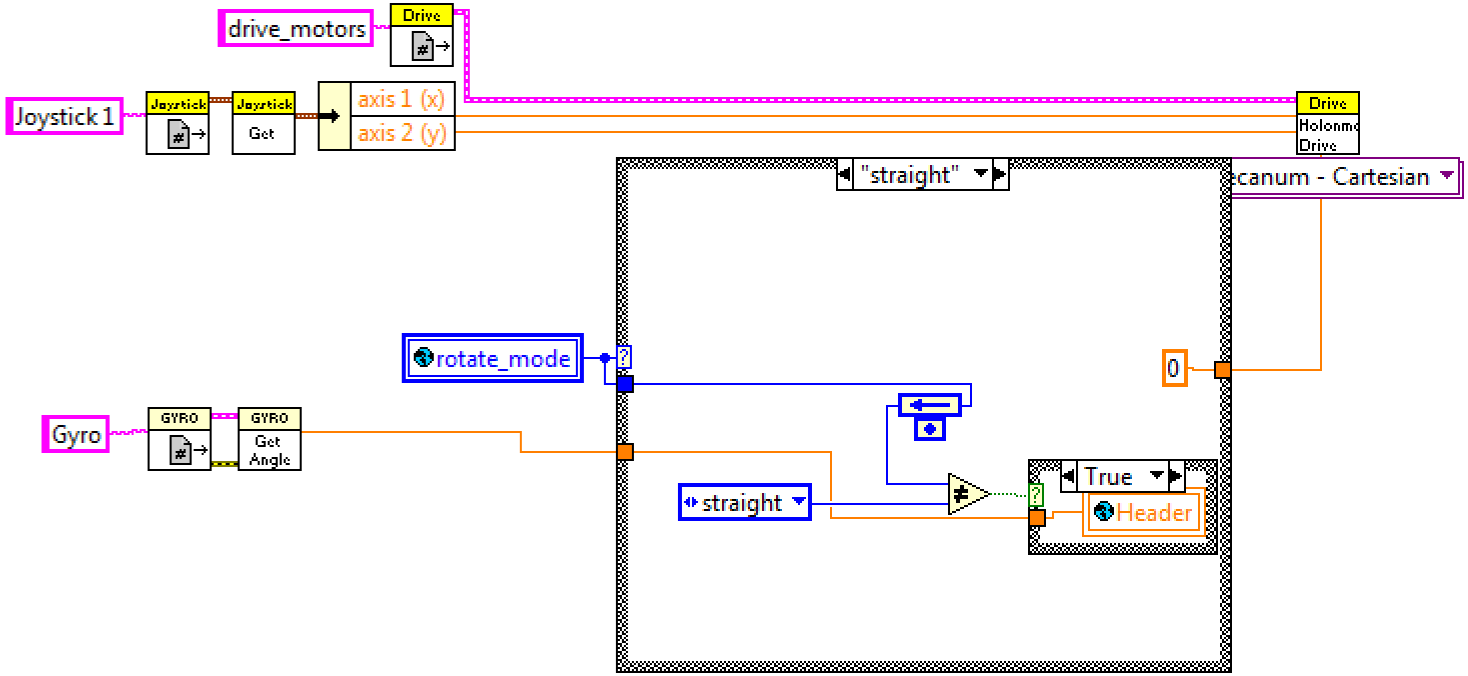
- The following code should be placed in Begin.vi to initialize the gyro



- You may need to change the gain based on your gyro-to do this, rotate the gyro 90 degrees and measure the output, and adjust the gain accordingly until the output is 90

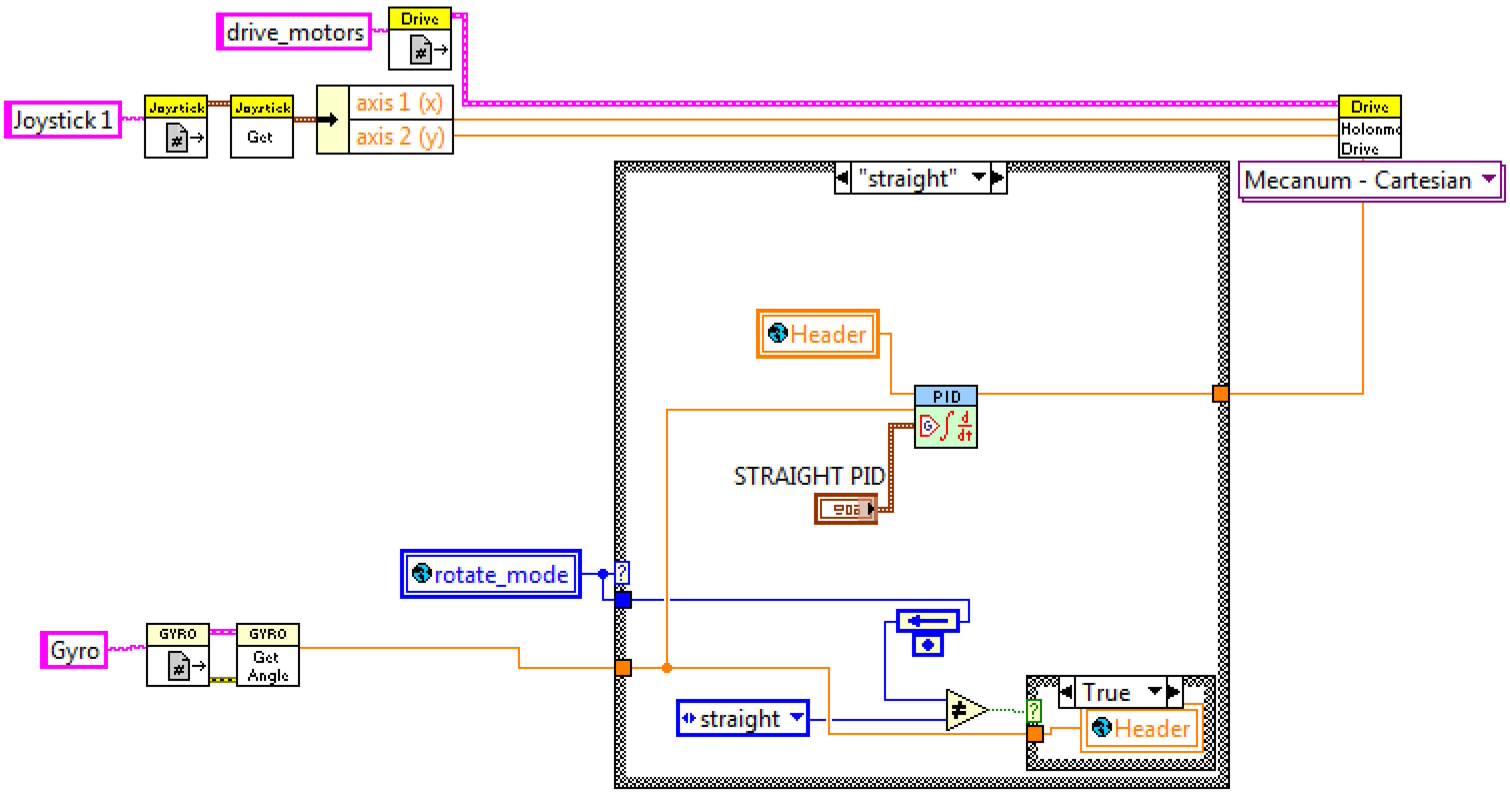
Step 6: Maintaining a fixed header

- The desired header is whatever direction the robot is facing the first time it is told to go straight
- This code (Teleop.vi) sends the current gyro angle to a global variable "header" the first time rotate mode equals "straight"



Step 6: Maintaining a fixed header

- Now that we know our header, we have to lock onto that
- We will implement LabVIEW's PID VI to accomplish this:

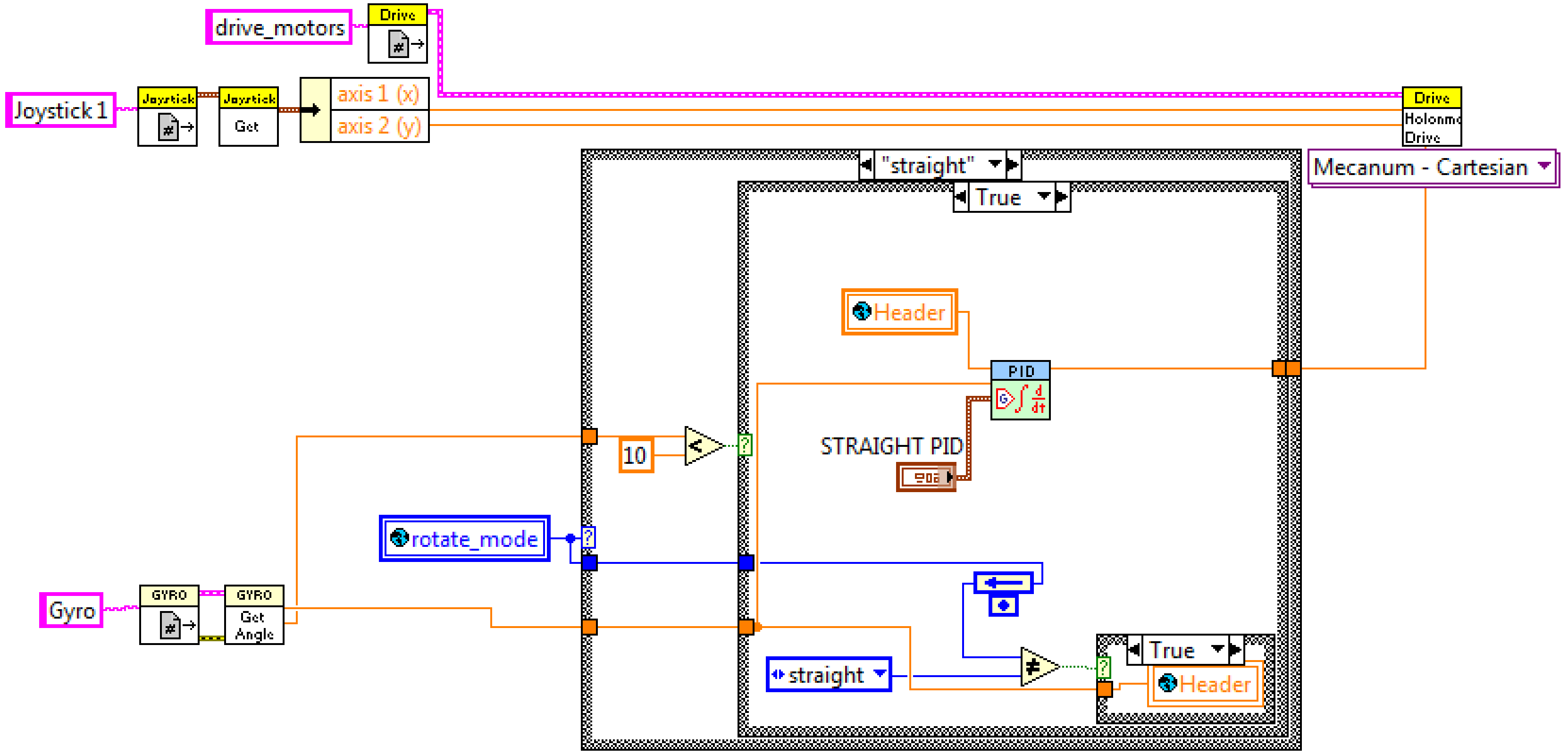


Step 6: Maintaining a fixed header

- To tune the PID loop, start with a proportional "P" gain of around 0.025
- You may find you don't need "I" or "D"
- Test your robot on the floor: you should be able to drive very smoothly
  - You might notice that after rotating, the robot will jerk back a little bit
    - This is because the robot still has some angular momentum in the transition from rotating to straight, and this code tries to compensate for that
    - Let's fix this...

Step 6: Maintaining a fixed header

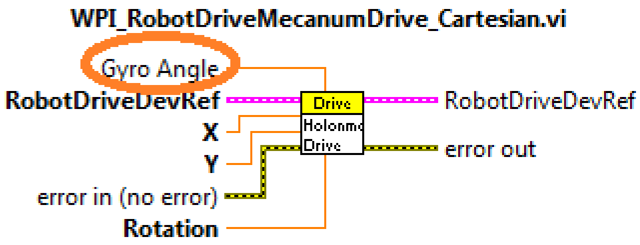
- Let's only try to go straight once the robot's angular rate is less than 10 degrees/second (this comes right out of the Gyro Get Angle VI)



- In the false case, send "0" to rotation

Step 7: Field-Oriented Drive

- When you push up on the joystick, it travels in whatever direction it's facing
  - This is "Robot-centric"
- Wouldn't it be better if the robot would simply move straight away from you if you push up?
  - This is "Field-centric", and is accomplished by rotating the joystick input using basic vector math
  - This is built right into the Holonomic VI: just wire in the gyro angle



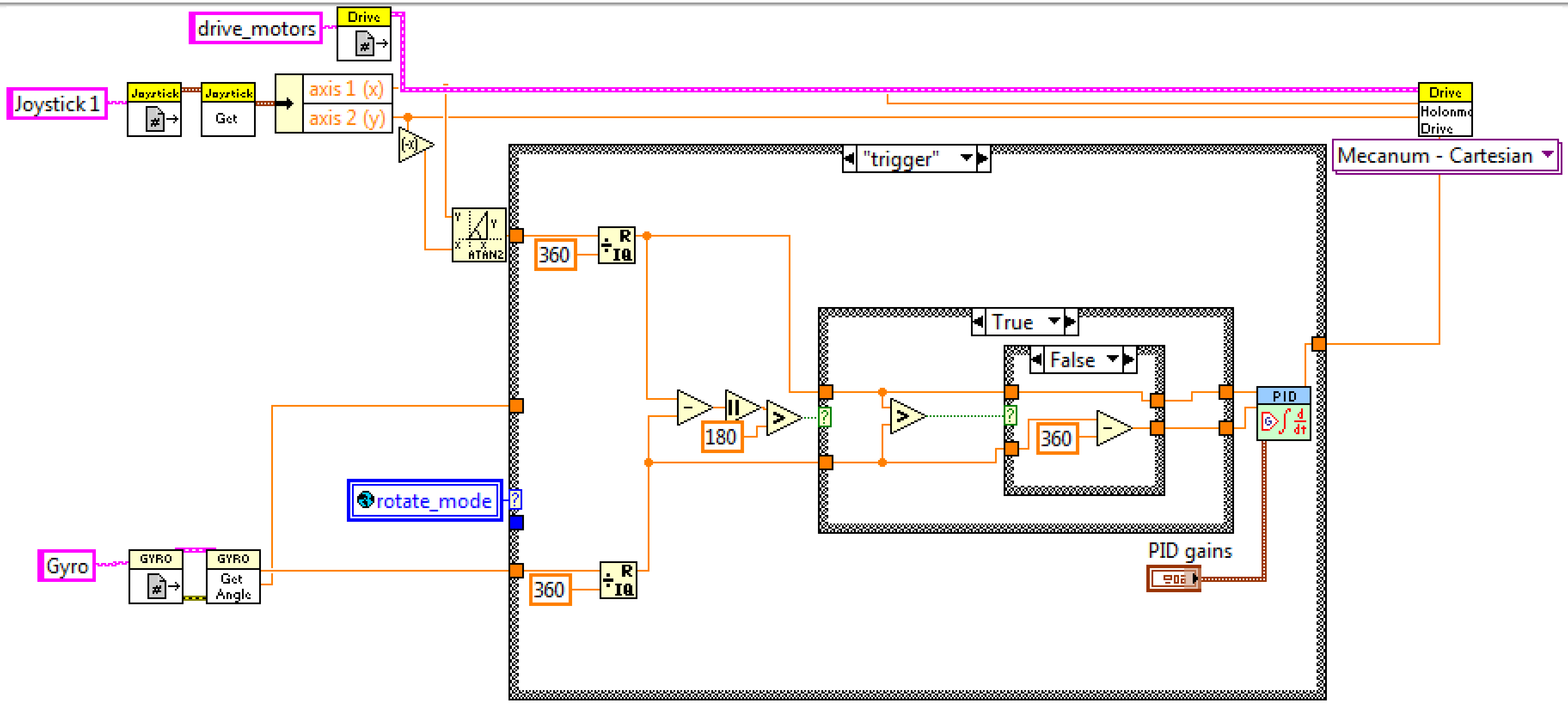
Step 8: A Recommended Extra

- Imagine that the robot is facing North, and you push the joystick right
  - The robot should maintain its North header while strafing to the right
  - Wouldn't it be convienient to--on the push of a button--rotate to the direction you're traveling?
    - To calculate this angle, do Math.atan2(Joystick X, -Joystick Y)
    - We will now implement a PID loop…

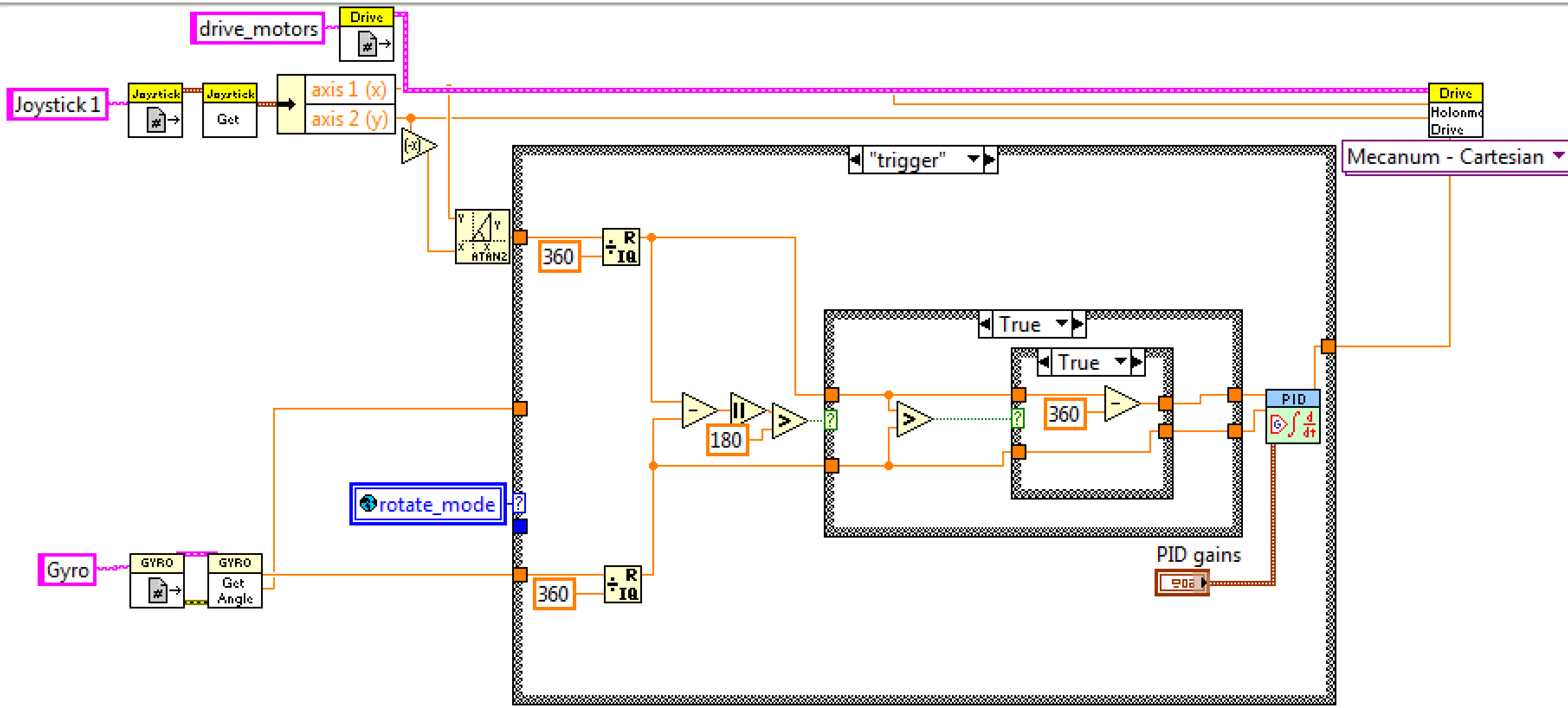
Step 8: A Recommended Extra

- Arithmetic is necessary to make sure that the robot doesn't try to correct for angles of, say, 480 degrees
  - First take both Joystick Angle and Gyro Angle and take them modulo 360
  - Subtract the two and take the absolute value
  - If the difference is greater than 180, subtract 360 from the larger, and enter the PID loop (with adjusted Joystick Angle as **setpoint**, and adjusted Gyro angle as **process variable**)
- An additional rotation mode is also necessary: we call this "trigger"

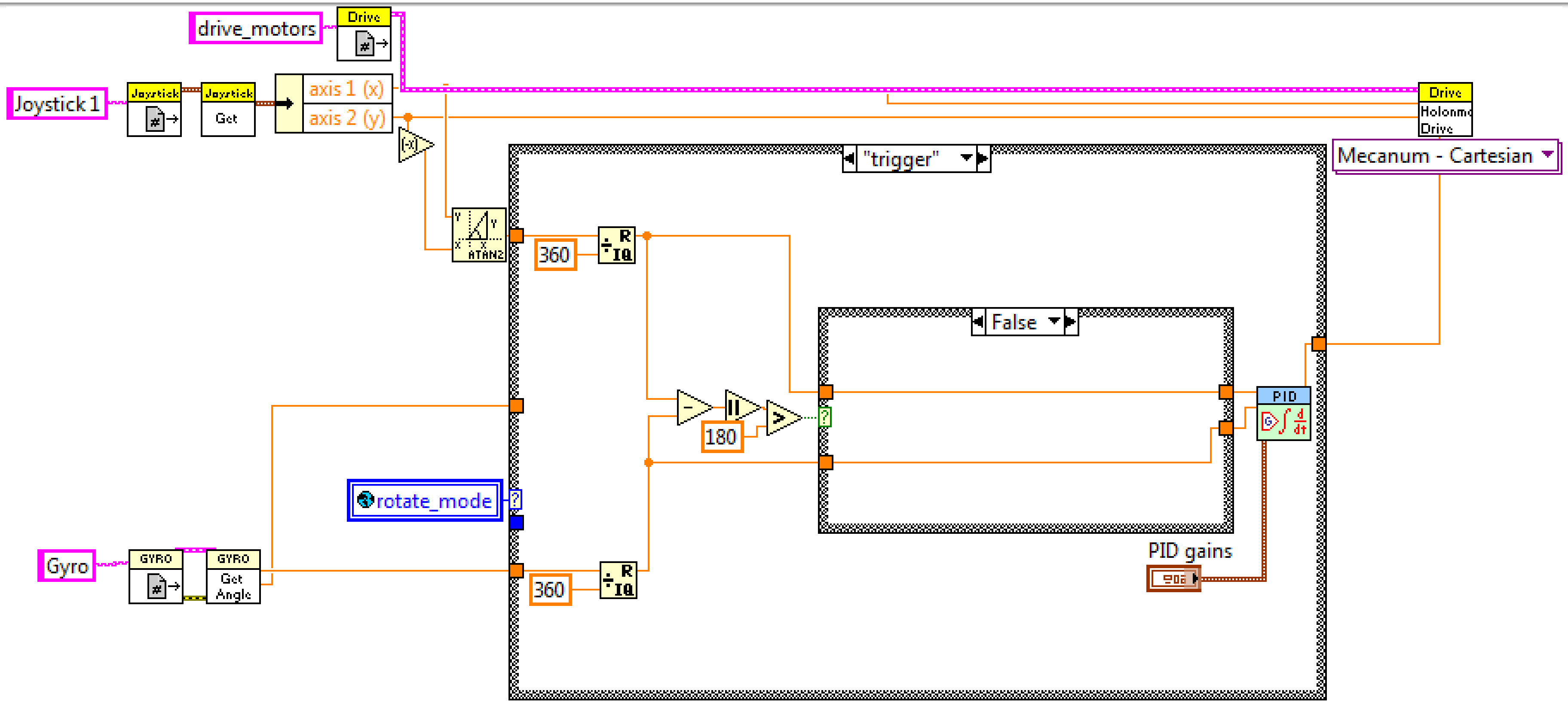
Step 8: A Recommended Extra



Step 8: A Recommended Extra



Step 8: A Recommended Extra



Related Links

- Inverse Kinematics Solution - the physics behind the mecanum drive calculations
- PID Theory Explained - more details about Proportional-Integral-Derivative control loops
- AndyMark - a great place to order your mecanum wheels

- Questions? Email webmaster@pobots.com