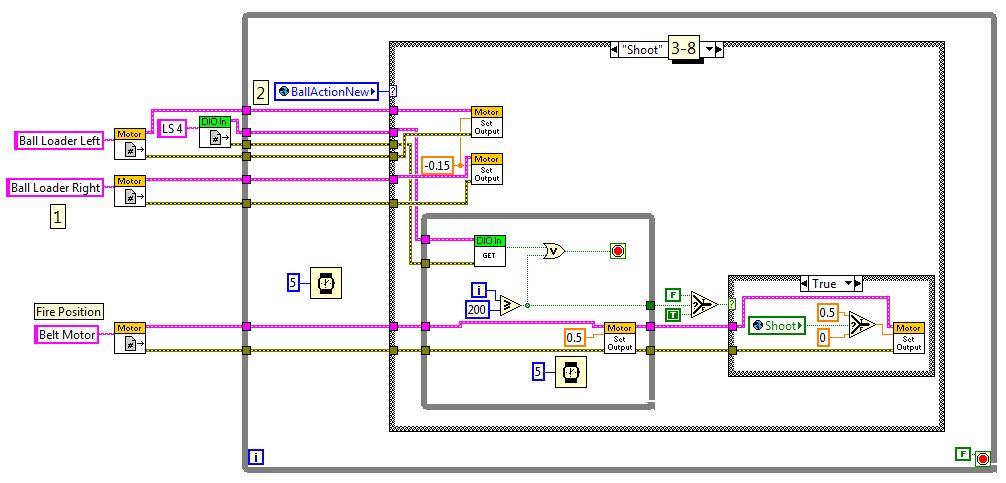
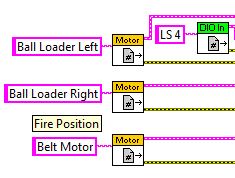
Overall View

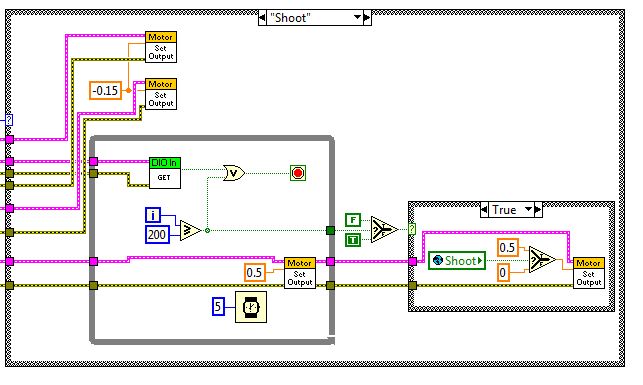
****

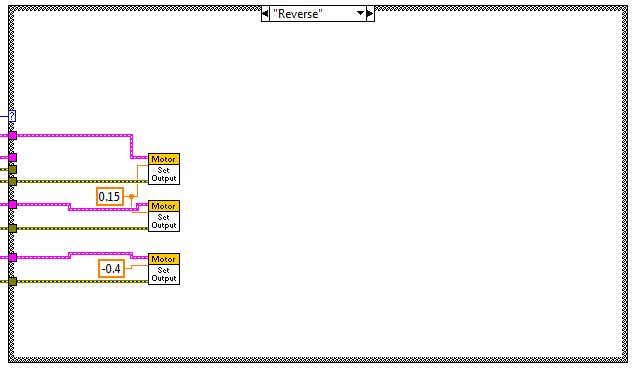
1. **References:** BallLoaderLeft, BallLoaderRight, BeltMotor (moves the ball within the robot), LS 4 (checks whether the ball is within the fire position). These motors/DIOs are used in the following cases.



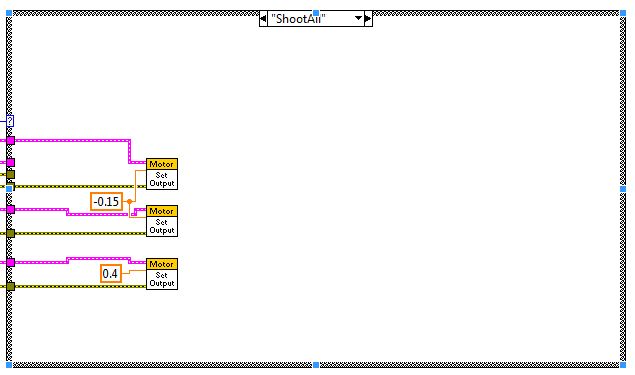
1. **Global Variable: BallActionNew:** Contains the options for the case structure, this variable is changed by the joystick commands in Teleop.VI

C:\Users\Feds201-2\Google Drive\Programming\Feds201 Manual\Images\BallLoaderNew.VI\BallActionNew.JPG

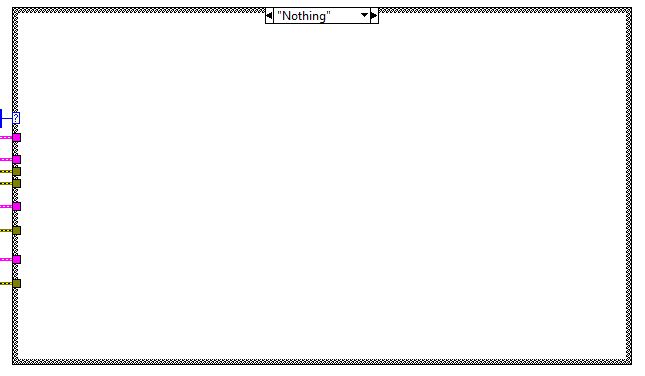
1. **Shoot Case:** BallLoader motors set to reverse to not accidentally get any extraneous balls into the robot; while loop makes the belt motor spin until 1 second has elapsed (200 iterations in inner loop X 5 millisecond delay on outer loop = 1000 ms total runtime), or the DIO has detected a ball in the “ready to shoot” position; once the loop or the 1 second completes, the final case structure waits for the final input from the user (shoot Boolean) and gives a final push on the belt motor to elevate the ball up to the spinning shooting wheel, at which point it shoots from the robot. 
2. **Reverse Case**: Reverses the belt motor in order to allow for the ball to move down within the robot when needed.



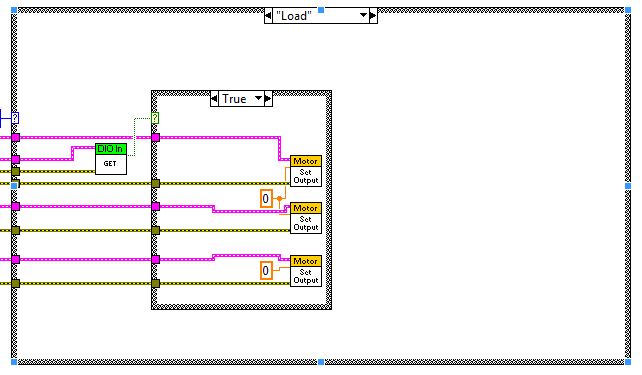
1. **ShootAll Case:** Extra case used to shoot all of the balls from the robot at once by simply running the belt motor forward.

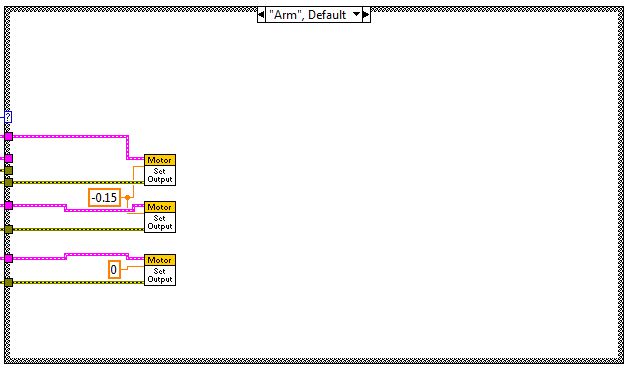


1. **Nothing Case:** Nothing. (fail-safe or to continue the robots current actions)



1. **Load Case:** Basically moves the belt up until the ball hits the DIO switch signifying it is in the loaded position. When it hits the DIO switch, stop the belt in order to prevent accidentally shooting the ball.



1. **Arm/Default** **Case:** stop the belt movement and put ballLoader motors on reverse – puts a robot in a type of “standby” mode. 

**Issues:**

* Timing of the balls loading into the robot was tough, balls wouldn’t stop at limit switches, delays were too long and wouldn’t catch the click of the switch.