Limit Switch Board Document

1 Introduction

This brief document provides a general overview of the Limit Switch Board (or Limits Board), designed for the University of Waterloo Robotics Team (Mars Rover 2018) by Josh Rubin. The schematic and layout for this board can be found on the UW Robotics GrabCad, or the reader can contact the designer to obtain the files if necessary. The reader can contact the board designer at josh.rubin@edu.uwaterloo.ca.

The board serves two major purposes on the UW Mars Rover: it acts as a hardware solution to monitor and control the mechanical limits of the rover's arm, and to automatically control a fan based on the ambient temperature.

2 The Board

The board consists of two major circuits; the limit switch combinational logic, and the temperature dependant fan controller. Each of these circuits will now be discussed in further detail.

2.1 Limit Switch Combinational Logic

This section of the board deals with handling the mechanical limits of the rover's arm. At a high level, the circuit cuts the PWM that is delivered to a motor controller when a limit switch has been triggered and the microcontroller is still supplying the direction signal in the direction that is no longer physically allowed. If this is not the case, the circuit passes PWM to to a motor controller. The circuit does not simply break the PWM indefinitely. If the opposite direction is provided by the controller, the PWM signal will be allowed to pass again, and once the arm reaches a location where the limit switch is no longer closed, the circuit will go back to its original state. The circuit's operation is clearly explained by an example in Section 2.1.1.

The circuit for 1 motor with bidirectional limiting (mechanical limits on two directions of motion; say, arm moving right and arm moving left) consists of 2 Single Pole Double Throw (SPDT) limit switches, a 2 input XOR gate, and a 2:1 multiplexer (MUX), and connections to HIGH (3.3V), LOW (GND) and associated pull-up and pull-down resistors.

Figure 1 below illustrates a high level (block diagram) of the circuit. The box on the left represents the microcontroller (located on the Arm Board) providing the direction and PWM signals to the motors, the middle block is the combinational logic, and the right block represents the output direction and PWM being fed to the motor drivers.

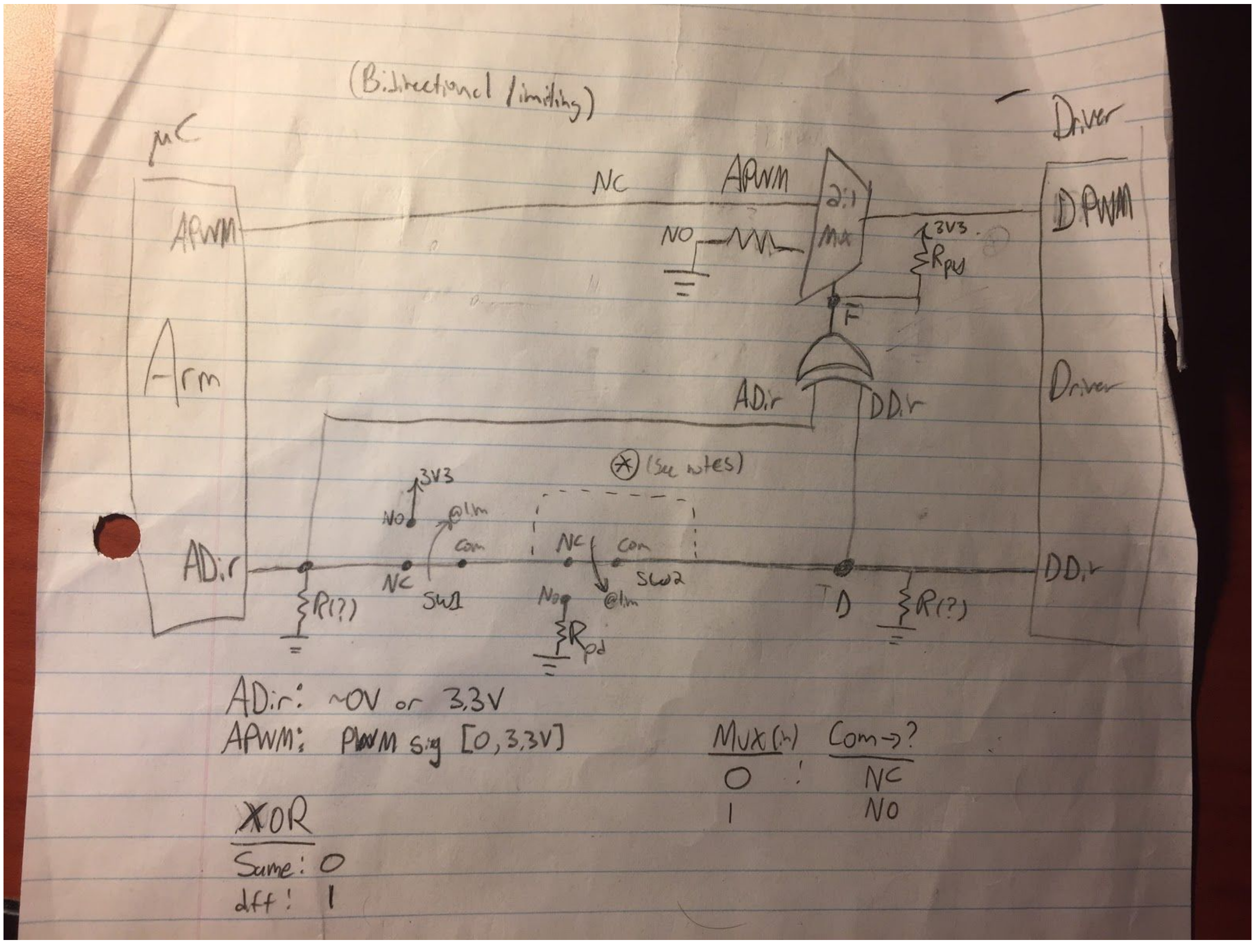

Figure 1. A block diagram of the limit switch combinational logic.

The operation of the circuit is best described through the use of an example.

2.1.1 Example of Operation

The limit switches are connected so that the direction input is the same as the direction output when a limit has not been reached (ADir = DDir). If one of the limits is reached, the limit switch then switches to the NO position, which is tied to either HIGH or LOW. Since the direction signal is either HIGH (defined as RIGHT for example) or LOW (defined as LEFT for example), this essentially asserts a direction on the motor controller. That is, when a limit switch is triggered, the circuit asserts the opposite direction (i.e., the direction the arm is allowed to move in) on the motor. The next paragraph discusses the PWM, which controls whether or not the motor is moving.

The XOR logic (LOW if the inputs are the same, HIGH if they are different) outputs the digital HIGH/LOW input of the selector on the MUX. If the two direction signals are the same, then the PWM signal will be selected by the MUX (NC signal). If the direction signals are different as is the case when a limit switch has been hit, GND (0 PWM) will be selected by the MUX. That means that when a limit switch has been triggered, the circuit will disable the PWM, and thus stop the motor from moving. The next paragraph discusses how the motor can move in the opposite direction again after a limit has been reached.

The circuit does not prevent the motor from moving indefinitely. If the microcontroller provides the same direction as the one being asserted by the limit switch (that is, the direction of valid motion), then the two direction inputs become the same and thus the XOR logic will ensure that PWM is selected. The reason this is possible is because the ADir and DDir line (seen in Figure 1) is essentially broken into two when a limit is reached. Thus, when a limit is reached, ADir and DDir become different lines, and are therefore independent. This is one of the reasons the logic works in the first place. Once the arm moves such that the limit switch is no longer closed, the circuit goes back to its original state. This logic therefore prevents the motor from being stuck indefinitely.

2.1.2 Notes

1. Shorting one of the limit switches in the circuit causes the limiting to become unidirectional (the operation is the same, except only one direction can be tracked and asserted).

2. The “R(?)” pull-down resistors in Figure 1 are implemented in the final design as pull-down resistors to ensure default logic values at the XOR gates.

3. Additionally, the pull-up resistor (versus pull-down) are used between the XOR and MUX since the sourced XOR uses an Open Collector Output.

4. The D and F markers on Figure 1 (near DDir and the output of the XOR gate) are two signals that are provided to the board’s microcontroller to indicate whether or not a limit has been reached (F) and which direction is currently being asserted by the circuit (D). Note that F could possibly be LOW (not asserted) even if one of the limits is switched. This would only be for the brief period of time between when the motor is moving in the correct direction but the limit switch is still triggered. That being said, if the limit switch is still activated and the controller again provides the invalid direction, the circuit will again cut the PWM since the XOR output will become HIGH once again. This edge case should not impact the operation or correctness of the monitoring.

2.1.3 Sourced Logic Components

2:1 MUX: http://www.ti.com/lit/ds/symlink/ts3a24159.pdf
XOR: http://www.ti.com/lit/ds/symlink/sn74lvc2g86.pdf

2.2 Fan Controller

The fan circuit uses the Microchip TC648 Fan Controller. This controller provides a PWM signal to control the speed of a fan depending on the output from a simple temperature sensor circuit.

Depending on the choice of thermistor and resistor values in the temperature circuit, the designer can adjust the temperature range in which the fan will operate, and that PWM will be supplied to the fan within that range. For example, in the originally designed circuit for this board, the fan will run at 50% PWM at 30 degrees, 100% PWM above 50 degrees, and have a roughly linear increase in PWM over that temperature range. The fan will also automatically shut off when the temperature is below 30 degrees (based on the output of a simple voltage divider circuit).

The circuit is essentially the same as the one shown in the device datasheet. The datasheet can be found here: http://ww1.microchip.com/downloads/en/DeviceDoc/21448D.pdf.