**Traffic Light by Valentin Vasilev**

The Hardware I’ve used for this assignment is as follows:

1xBreadboard 860 pins

2x RED LEDs

2x GREEN LEDs

4x 370 Ohm resistors

4x 110 Ohm resistors

Jumper Wires

1x Arduino Nano (Old Bootloader)

1xArduino Uno

1xCapacitor touch sensor 8 pads

The software used:

Arduino IDE

SoftwareSerial library

Since I wanted to create this project as close as possible to the real world I decided to use a 8pad touch capacitor sensor as a “Control Panel” for the Master MCU (Nano). The touch sensor can be swapped with 8 physical buttons but debouncing will have to be done. The resistors are connected in series right before each LED (110 Ohm in series with 370 Ohm which gives is a total resistance of 480 Ohms, close enough to the one requested in the assignment).

The Arduino Nano is the Master in my communication protocol. It has the control panel attached and configures the traffic light’s mode. The Arduino Uno is the Slave.

The protocol is constructed as follows:

Start byte - ‘<’ | Then is followed by the command - “mode” | followed by a separating byte - ‘=’ | Then comes the command’s value - ‘1-6’ (an integer between 1 and 6) | Final closing byte - ‘>’

Example: <mode=4>

In the protocol is implemented a response message which the Slave returns to the Master if it understood the message, which is “<OK>”. If the Master doesn’t receive an OK response from the Slave after trying to reach it 5 times, it considers the Slave disconnected.

If a message is not understood by the Slave, an “<OK>” is not sent back, so the Master sends its command 4 more times until understood.

Everything works as intended under normal conditions. Under heavy load as well (If I spam commands and press the buttons/sensors repeatedly). If you disconnect the communication the Slave will operate the Traffic Light under the last given instruction (turned OFF by default)..

Extra functionality:

6 modes:

Button 1 - MASTER START with GREEN, hold GREEN 2 seconds continue RED & cycle

SLAVE START with RED, hold RED 2 seconds continue GREEN & cycle

Button 2 - MASTER & SLAVE switch to RED until other mode

Button 3 - MASTER & SLAVE switch to GREEN until other mode

Button 4 - MASTER START with GREEN, hold GREEN 5 seconds continue RED & cycle

SLAVE START with RED, hold RED 5 seconds continue GREEN & cycle

Button 5 - MASTER START with GREEN, hold GREEN 5 seconds continue RED & cycle

SLAVE START with RED, hold RED 5 seconds continue GREEN & cycle

Button 6 - BOTH MASTER & SLAVE turn off ALL lights.

Button 7 & 8 – Not implemented yet (can be implemented easily due to the protocols easy integration), can be added sensors and based on the sensors’ values adapt to the traffic and so on.