

### **Possible protocol changes to simplify project integration:**

Removed all the messages sent from the traffic lights. Now traffic lights will just answer to controller requests. Status(X) will be the response from the traffic light when the controller do a Ping(x) request. The value which was defined to be send in TIME(x) now is obtained by the controller in the Status(X) message. The traffic lights will never do Ping(x) to other traffic lights neither to controller. The ACK (x) should be send as response to RED(x), GREEN(x) and OFF(x) requests.

### **Code to communicate:**

At the setup every traffic light should do `wire.begin(entryNumber)` where `#entryNumber` is defined by the wires configuration.

```
void setup() {  
    Int entryNumber =getEntryNumber();  
    Wire.begin(entryNumber);    // join i2c bus with address #entryNumber  
    Wire.onReceive(receiveEvent); // when master sends a message  
    Wire.onRequest(requestEvent); //when master asks for the answer  
}
```

When to answer a request the traffic light should do:

```
void requestEvent(){  
    Wire.write(answer);  
}
```

When the controller wants to send a message he should do:

```
void sendMessage(){  
    Wire.beginTransaction(entryNumber); // transmit to device #entryNumber  
    Wire.write(message);                // sends the message  
    Wire.endTransmission();  
    Wire.requestFrom(entryNumber, x); // request x bytes from slave device #entryNumber  
    while (Wire.available()) {          // slave may send less than requested  
        char c = Wire.read();           // receive a byte as character  
        Serial.print(c);                // print the character  
    }  
}
```

All the messages except the Status(x) should have the following format:

Sender|OperationNumber|Destination|Integrity Byte. The Status should be

Sender|OperationNumber|Destination|Information|Integrity. Each one of these members is **1 byte** long so, **they must be converted to char**.

1. The sender will always be the controller (entry 0) but it is maintained for possible future needs, for the same reason the Destination was kept.
2. The operation numbers are:
  - RED(x)->0
  - GREEN(x)->1
  - OFF(x)->2
  - PING(x)->3
  - ACK(x)->4
  - STATUS(x)->5
3. The Integrity byte is the char of the sum of the previous message bytes: char(Sender+OperationNumber+Destination+Message)
4. Information- the meaning of the 8 bits is [ pedestRedFailing, pedestYellowFailing, pedestGreenFailing, redFailing, yellowFailing, greenFailing, timerActivated, 0], the first 3 bits should be 1 if the lights near pedestrian are broken, the following 3 should be 1 if the other traffic lights are broken, the seventh bit should be 1 if the pedestrians pressed the button, the last bit has no meaning.

### **RED(x)**

The message sent from the controller to traffic light #x will be:

0|0|x|x and the response which is an ACK message will be x|4|0|x+4

### **GREEN(x)**

The message sent from the controller to traffic light #1 will be:

0|1|x|1+x and the response, which is an ACK message, will be x|4|0|x+4

### **OFF(x)**

The message sent from the controller to traffic light #1 will be:

0|2|x|2+x and the response, which is an ACK message, will be x|4|0|x+4

### **PING(x)**

The message sent from the controller to traffic light #1 will be:

0|3|x|3+x and the response, which is an STATUS message, will be x|5|0|Status|x+Status+5

Useful sources:

Multi-master:

<http://forum.arduino.cc/index.php?topic=13579.0>

Master working as proxy:

<https://www.arduino.cc/en/Tutorial/MasterReader>

Wire library:

<https://www.arduino.cc/en/Reference/Wire>