Overview - For this project I built an Onramp Metering System using Arduino and Processing. The Microcontroller used was Arduino (based on Atmel168) and the programming languages were Processing and Wiring (derivative of Processing used for hardware). For the implementation I used two pushbuttons as the sensors and 3 LED’s that would serve as the stop, go, and a warning signal. With processing, I implemented a heads up display for the road conditions.

Implementation
For this Metering System, I used two sensors – sensor 1 would keep track of the cars that are currently on the freeway, while the second sensor would keep track of the cars entering the ramp. Depending on the values of these two counts the Arduino Microcontroller will make a decision as to the rate of entry. The pseudo code for this process is given below. The first case, where both count variables are zero, is a “don’t care” state since there are no vehicles entering the onramp and there are no cars on the freeway. The second case is when there are cars on the freeway but there are no cars on the on ramp. Green is displayed because, otherwise, the car entering the onramp would have to wait up to 16 seconds for the green light to turn on. The third case is similar to the second; the only difference is that there are no cars on the freeway and some cars on the onramp. The last case is the most important – from the pseudo code - you can see how the rate of entry will be calculated. As the numbers of cars on the freeway and on the on ramp increase the slower the rate of entry will be. Eventually, if there are too many cars then the rate of entry will decrease and a warning signal will be generated - this will inform drivers of heavy congestion and slow traffic speed.

Psuedo-Code

```java
If(sensor 1 count == 0 and sensor 2 count == 0)
    display red (green can also be used in this case)
Else if(sensor 1 count > 0 && sensor 2 count == 0)
    display green
else if(sensor 1 count == 0 && sensor 2 count > 0)
    display green
else if(sensor 1 count > 0 && sensor 2 count > 0)
{   if(sensor 1 count < 7)
    {
```
if(sensor 2 cnt < 5)
  cars will enter every half second
else
  cars will enter every 1 second.
}
else if(sensor 1 cnt >= 7)
{
  if(sensor 2 cnt < 5)
    cars will enter every 2 seconds
  else
    cars will enter every 3 seconds and a warning signal will light up
    (the warning signal will alert drivers of heavy congestion)
}

In this implementation, the entry signal will be generated every 16 seconds, we could certainly decrease the time interval to increase the precision of the metering system, however, in this case 16 seconds was chosen because it provided enough time to push the pushbuttons.

In the initial stages of the project, I had both counters outside of the microcontroller (I used and SN74LS93 Chip) since there would be concurrency issues with the program and the signals - the microcontroller would need to keep outputting a signal while the counts were inputted from the sensors. After doing research on this, I found that it would be easier to implement the sensor inputs to the microcontroller using interrupts. In Arduino, they are easily implemented, as they only require one statement to initialize and a function that is called when the interrupt occurs (see code and references for further explanation).

After solving the timing issues using interrupts, I moved onto Processing. As previously mentioned Processing is used to provide drivers with the freeway conditions. This is implemented by sending a serial signal (in this case an integer) to Arduino to Processing via the USB cable. Once the signal is processed then an appropriate message will be displayed - Ideally the message would be sent to an LCD display, but I wasn’t able to implement that feature for this project due to time constraints. Instead the message is only displayed on the computer screen.

Note: In the real world, the push button sensors could be replaced with ultrasonic sensors (for presence detection), pressure tubes, or some other sensor. For this project, the pushbuttons sufficed since they only served to simulate the traffic on the roads.

References
5.
# Arduino Code

```c
#include <avr/io.h>
#include <avr/interrupt.h>

int s1_cnt = 0;  // count variables
int s2_cnt = 0;
int warn = 0;
boolean pulseSign = true;
unsigned long init_time = 0;

int sensor1 = 2; //sense pins for interrupts
int sensor2 = 3;

volatile int val1 = 0;
volatile int val2 = 0;

int red = 13;  // output signal - red
int green = 12; // output signal - green
int yellow = 11;
int re_set = 10;

int delayT = 0;  // delay time
int startTime = 0;

void setup()
{
    pinMode(red, OUTPUT);
    pinMode(green, OUTPUT);
    pinMode(yellow, OUTPUT);
    pinMode(re_set, OUTPUT);

    pinMode(sensor1, INPUT);
    pinMode(sensor2, INPUT);

    attachInterrupt(0, func1, FALLING);
    attachInterrupt(1, func2, FALLING);

    digitalWrite(red, LOW);
    digitalWrite(green, LOW);
    digitalWrite(yellow, LOW);

    Serial.begin(9600);
}

void loop()
{
    init_time = millis() + 16000.;
    calculateDelays();
    pulseSign = true;
    resetFunc();
    sendSignal();

    while(pulseSign)
    {
        if(warn == 1)
        {
            digitalWrite(yellow, HIGH);
        }
        if(delayT > 0)
        {
            digitalWrite(green, HIGH);
            delay(1000);
        }
    }
    delay(200);
}
```

digitalWrite(green, LOW);
delay(delayT);
}
if(millis() >= init_time) // checks if 16s have elapsed
{
    warn = 0;
pulseSign = false;
    init_time = 0;
}
// reset signal - used for debugging
digitalWrite(re_set, LOW);
delay(1000);
digitalWrite(re_set, HIGH);
delay(1000);
digitalWrite(re_set, LOW);
delayT = 0;
}
// this function sends the signal via serial
void sendSignal()
{
    if(delayT == 0)
    {
        Serial.print(1);
    }
    else if(delayT == 500)
    {
        Serial.print(2);
    }
    else if(delayT == 1000)
    {
        Serial.print(3);
    }
    else if(delayT == 2000)
    {
        Serial.print(4);
    }
    else if(delayT == 3000)
    {
        Serial.print(5);
    }
}
// this function calculates the delays
// based on the sensor counts
void calculateDelays()
{
    digitalWrite(red, LOW);
digitalWrite(green, LOW);
digitalWrite(yellow, LOW);
    if(s1_cnt == 0 & & s2_cnt == 0)
    {
        digitalWrite(red, HIGH);
    }
    else if(s1_cnt == 0 & & s2_cnt > 0)
    {
        digitalWrite(red, LOW);
digitalWrite(green, HIGH);
    }
    else if(s1_cnt > 0 & & s2_cnt == 0)
    {
        digitalWrite(green, LOW);
digitalWrite(red, HIGH);
    }
    else
    {
        digitalWrite(red, LOW);
    }
digitalWrite(green, LOW);
digitalWrite(yellow, LOW);
if(s1_cnt < 7)
{
  if(s2_cnt < 5)
  {
    delayT = 500;
  }
  else // >= 5
  {
    delayT = 1000;
  }
  }
else if(s1_cnt >= 7)
{
  if(s2_cnt < 5)
  {
    delayT = 2000;
  }
  else // >= 5
  {
    delayT = 3000;
    warn = 1;
  }
}
// resets the count variables
void resetFunc()
{
  s1_cnt = 0;
  s2_cnt = 0;
}
//INTERRUPT 0 = read from pin 2
void func1()
{
  s1_cnt++;
}
//INTERRUPT 1 = read from pin 3
void func2()
{
  s2_cnt++;
}

Processing Code
import processing.serial.*;
import cc.arduino.*;

//INTERRUPT 0 = read from pin 2
void func1()
{
  s1_cnt++;
}
//INTERRUPT 1 = read from pin 3
void func2()
{
  s2_cnt++;
}

//INTERRUPT 0 = read from pin 2
void func1()
{
  s1_cnt++;
}
void func2()
{
  s2_cnt++;
}

Serial port;
PFont fontA;
// messages
String messg0 = " no traffic \n";
String messg1 = " no oncoming traffic \n";
String messg2 = " minimal traffic \n drive safe \n";
String messg3 = " traffic ahead \n drive safe \n";
String messg4 = " heavy traffic ahead \n proceed with caution \n";
int value;

void setup()
{
  size(600, 200); // window size
  noStroke();
  frameRate(1);
  println(Serial.list());
}
fontA = loadFont("HelveticaNeue-48.vlw"); // sets the font
textFont (fontA, 20); // sets the size

port = new Serial(this, Serial.list()[0], 9600);
}

do what}{

while(port.available() > 0) // checks if there's a signal in the buffer
{
value = port.read(); // reads in the value
outputMessage(); // calls the function to output a message
}

value = value - 48; // since the integer is sent via ascii we have to subtract by 48
if(value > 0)
{
if(value == 1)
{
text(messg0, 20, 50);
} else if(value == 2)
{
text(messg1, 20, 50);
} else if(value == 3)
{
text(messg2, 20, 50);
} else if(value == 4)
{
text(messg3, 20, 50);
} else if(value == 5)
{
text(messg4, 20, 50);
}
}