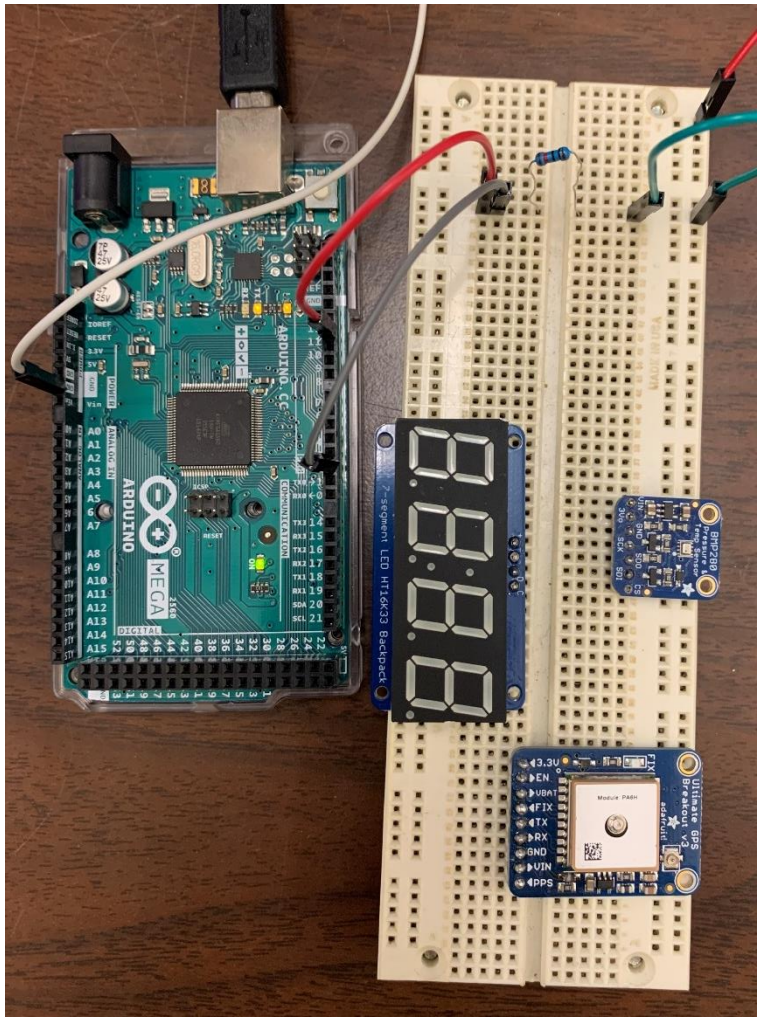
Cosmic Ray Data Acquisition Project

Jun ha Kim
Mentor: Professor Raul Armendariz
Queensborough Community College
Department: Physics

Arduino Mega and Breadboard Setup (Square Pulse Plot)



- Arduino Mega is connected to a solderless breadboard like so:
  - Arduino GND to negative (breadboard) power rail
  - Arduino Pin 11 to A10
  - Arduino Pin 2 to B10
  - Resistor connects E10 to F10 so that the voltage on Pin 2 does not float
  - J 10 to negative power rail
- Objective is to capture an electrical signal—Pulse Per Second (PPS) — that repeats once per second using the Ardunio Board

Code used to collect Pulse Point Data

sketch_apr01a

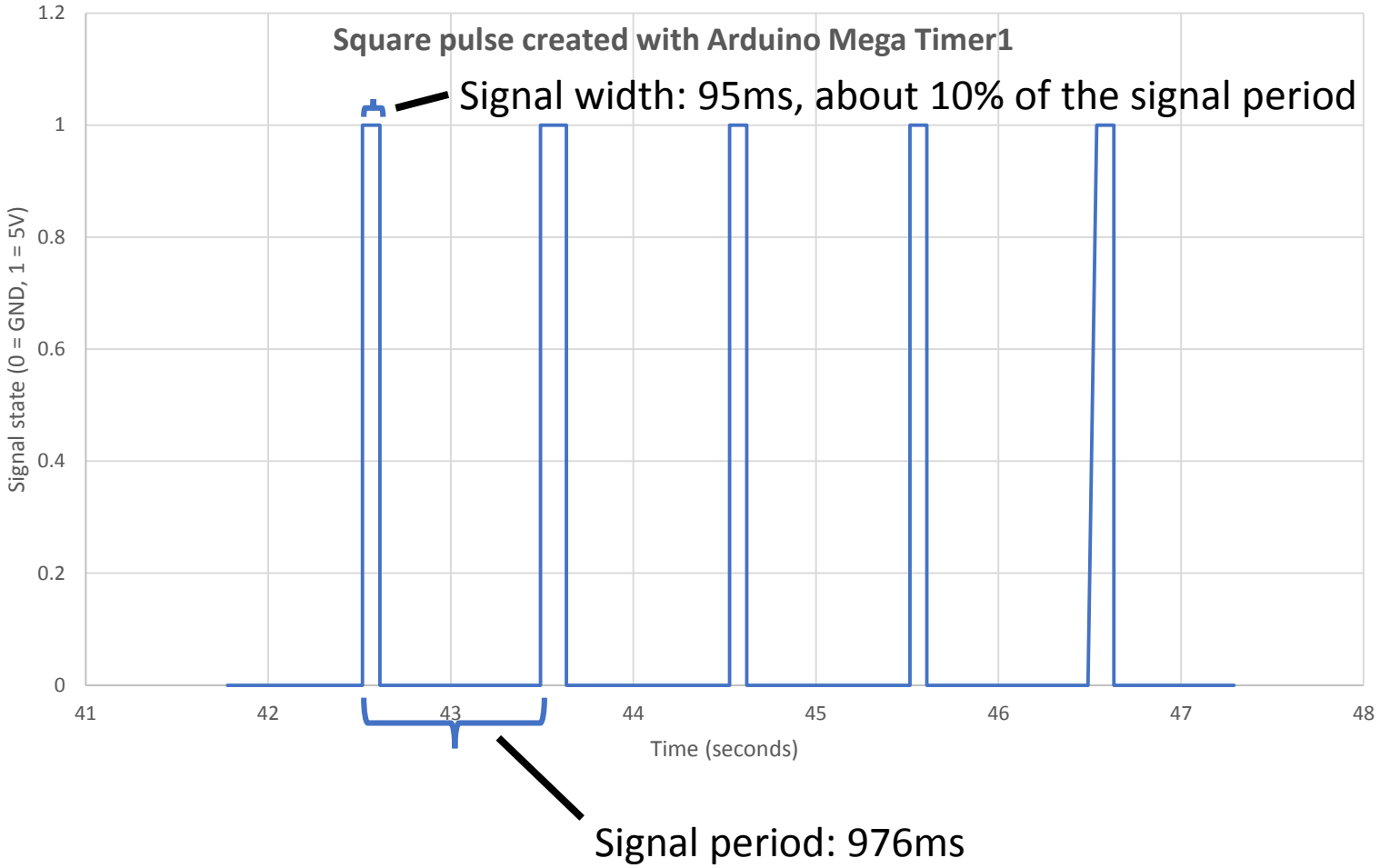
```
#include <TimerOne.h>

void setup() {
  Serial.begin(9600);
  pinMode(11, OUTPUT);
  pinMode(2, INPUT);
  Timer1.initialize(1000000);
  Timer1.pwm(11,100000);
}

void loop() {
  while(digitalRead(2) == HIGH) {
    Serial.println(HIGH);
  }
  while(digitalRead(2) == LOW) {
    Serial.println(LOW);
  }
}
```

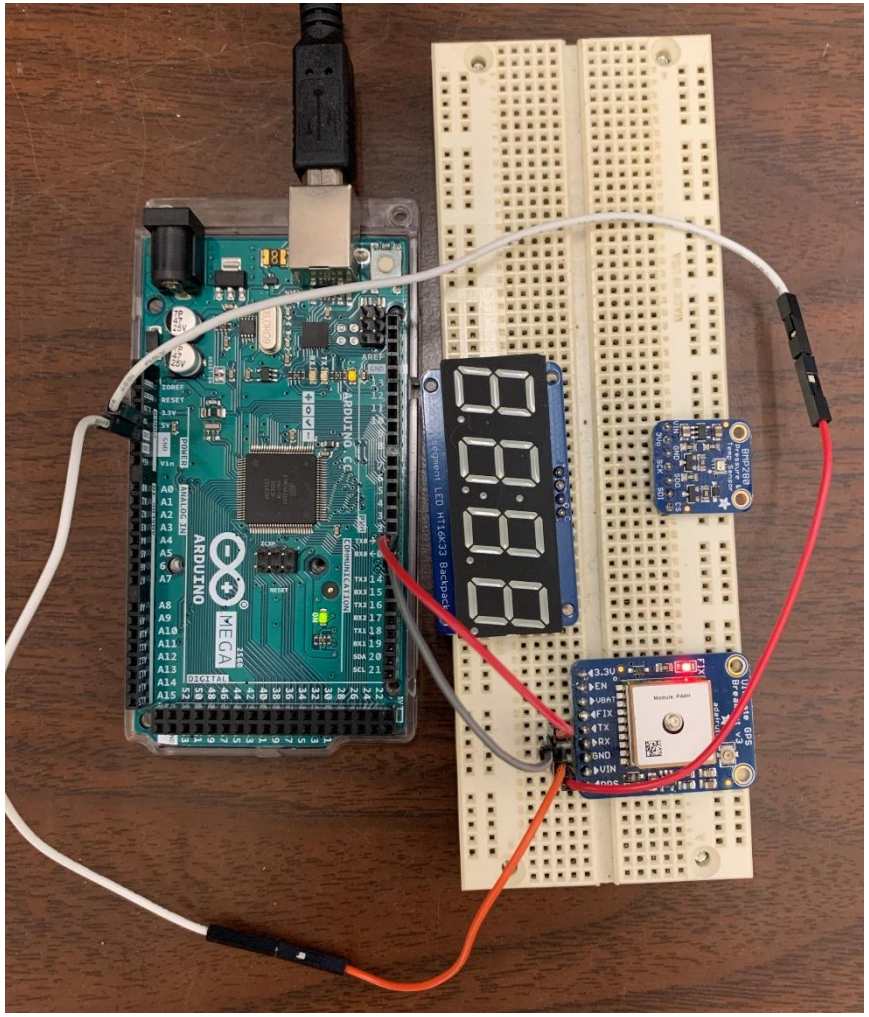
- Timer1.initialize() and Timer1.pwm() are used to generate a square pulse from pin 11 (Arduino board)
- Pulse has a 1 second period and 0.1 second width
- Loop() function ensures that:
  - If the input is high, the program prints a high number
  - If the input is low, the program prints a low number

Square Pulse Plot Results



- Signal goes high (1) at: 42.516 (SS:MS)
- Signal goes low (0) at: 42.611
- Signal goes high again at: 43.492
- Signal width: high – low = 42.611 – 42.516 = 95ms
- Signal period: $high_2 - high_1 = 43.492 - 42.516 = 976ms$

Arduino Mega and Breadboard Setup (NMEA data)



- Arduino Mega is connected to GPS receiver (placed on breadboard) like so:
  - Arduino 5V to VIN
  - Arduino GND to GPS GND
  - Arduino RX0 to GPS RX
  - Arduino TX0 to GPS TX

- National Marine Electronic Association (NMEA) data collected

Code used to collect NMEA Data

sketch_mar25a

```
1 void setup() {
2   // put your setup code here, to run once:
3 }
4
5 void loop() {
6   // put your main code here, to run repeatedly:
7 }
```

- Default baud rate (rate at which bits are transmitted) or 9600 bauds selected

GPS receiver NMEA data (sample)

- 11:17:39.939 -> $GPGGA,151739.000,4045.3472,N,07345.5606,W,2,09,0.90,207.5,M,-34.3,M,0000,0000*5B
- 11:17:40.033 -> $GPGSA,A,3,04,21,27,16,30,14,07,08,09,,,,1.67,0.90,1.40*0B
- 11:17:40.080 -> $GPGSV,3,1,11,08,83,088,28,07,59,307,46,27,46,049,28,30,32,310,49*72
- 11:17:40.174 -> $GPGSV,3,2,11,51,31,225,47,21,30,143,24,09,30,225,49,16,22,074,27*7B
- 11:17:40.220 -> $GPGSV,3,3,11,04,14,193,40,14,07,273,47,01,07,168,*49
- 11:17:40.314 -> $GPRMC,151739.000,A,4045.3472,N,07345.5606,W,0.01,211.12,150422,,,D*76
- 11:17:40.361 -> $GPVTG,211.12,T,,M,0.01,N,0.02,K,D*3A

- The data was collected in an indoor setting, resulting in faulty information initially
- To fix this issue, the GPS module was connected to a satellite receiver using interlinked coax cables
- The satellite was placed outdoors, pointing towards the sky

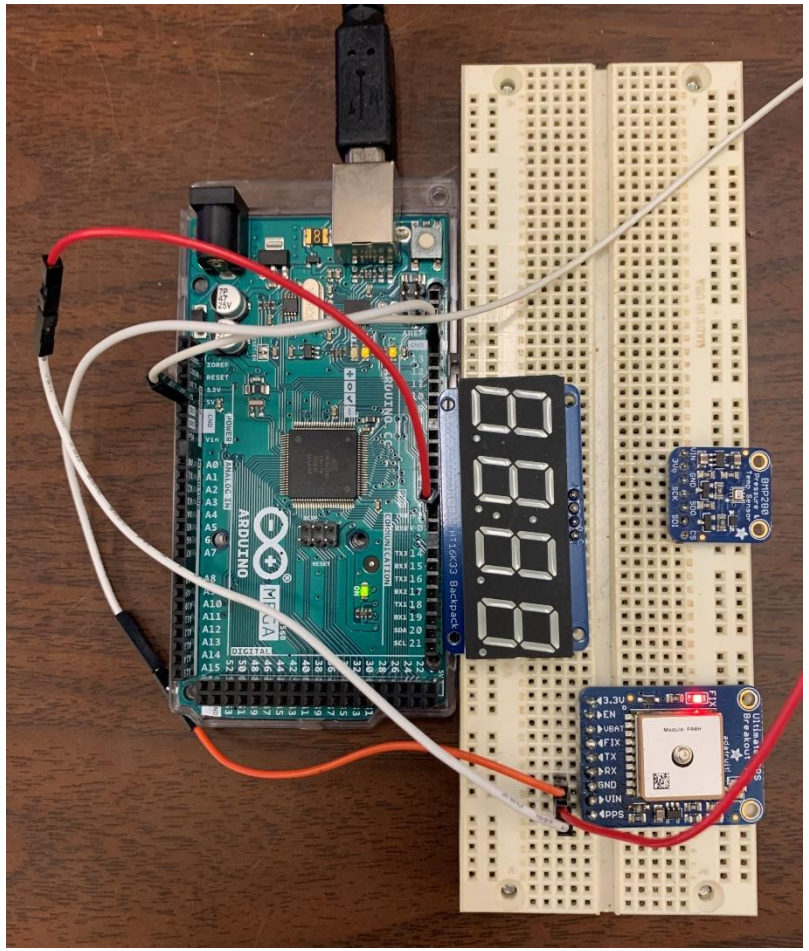
Understanding NMEA (National Marine Electronics Association) sentences

- All sentences begin with a "$" symbol
- Limited to 80 characters (including a "newline” character)
- Commas delineate distinct subjects (latitude, time, etc.)
- Sentences provide information regarding location, satellites the GPS successfully interface with, altitude above mean sea level, etc.
- Difference sentences may repeat identical info, but will also supply new, relevant information

Understanding NMEA sentences (Continued)

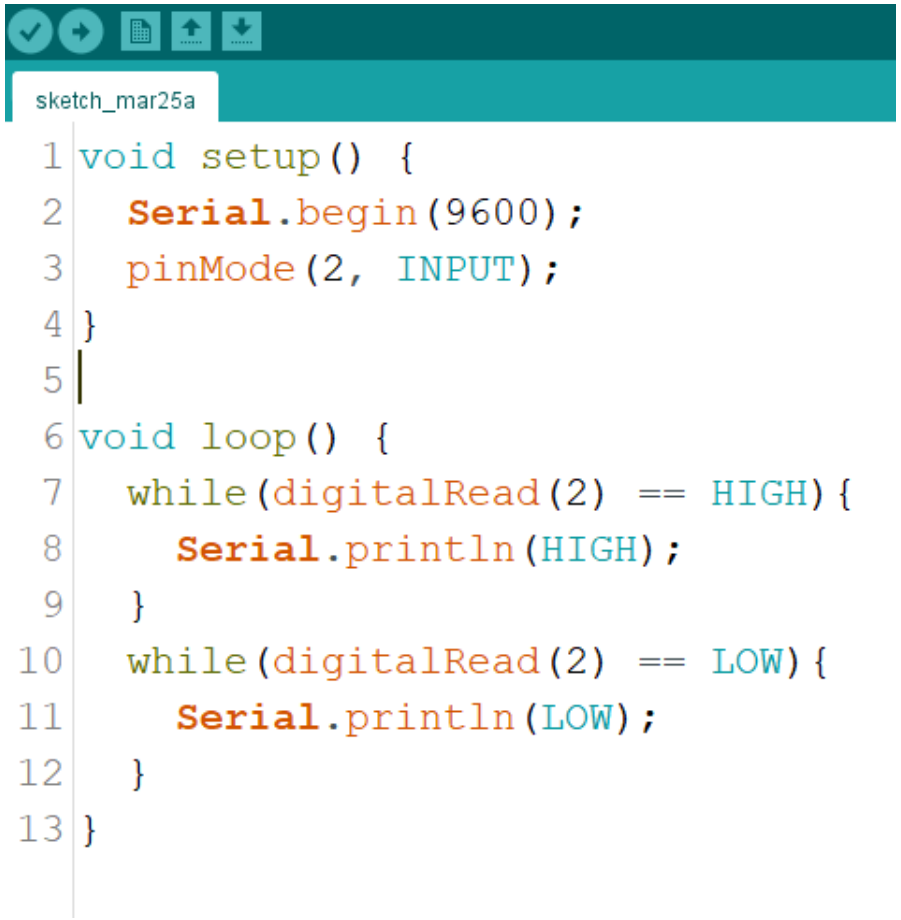
- $GPGGA,151739.000,4045.3472,N,07345.5606,W,2,09,0.90,207.5,M,-34.3,M,0000,0000*5B
  - GGA: indicates the data type and describes how the sentence should be interpreted
  - 151739: Time (UTC)
  - 4045.3472,N: Latitude 40 deg 45.3472’ N
  - 07345.5606,W: Longitude 73 deg 45.5606’ W
  - 2: Denotes fix quality
  - 09: Number of satellites being tracked
  - 0.9: Horizontal dilution of position
  - 207.5,M: Altitude above mean sea level (meters)
  - 34.3,M: Height of geoid (mean sea level) above WGS84 ellipsoid (meters)
  - *5B: Checksum data

Arduino Mega and Breadboard Setup (GPS Receiver PPS [Pulse Per Second] Square Pulse)



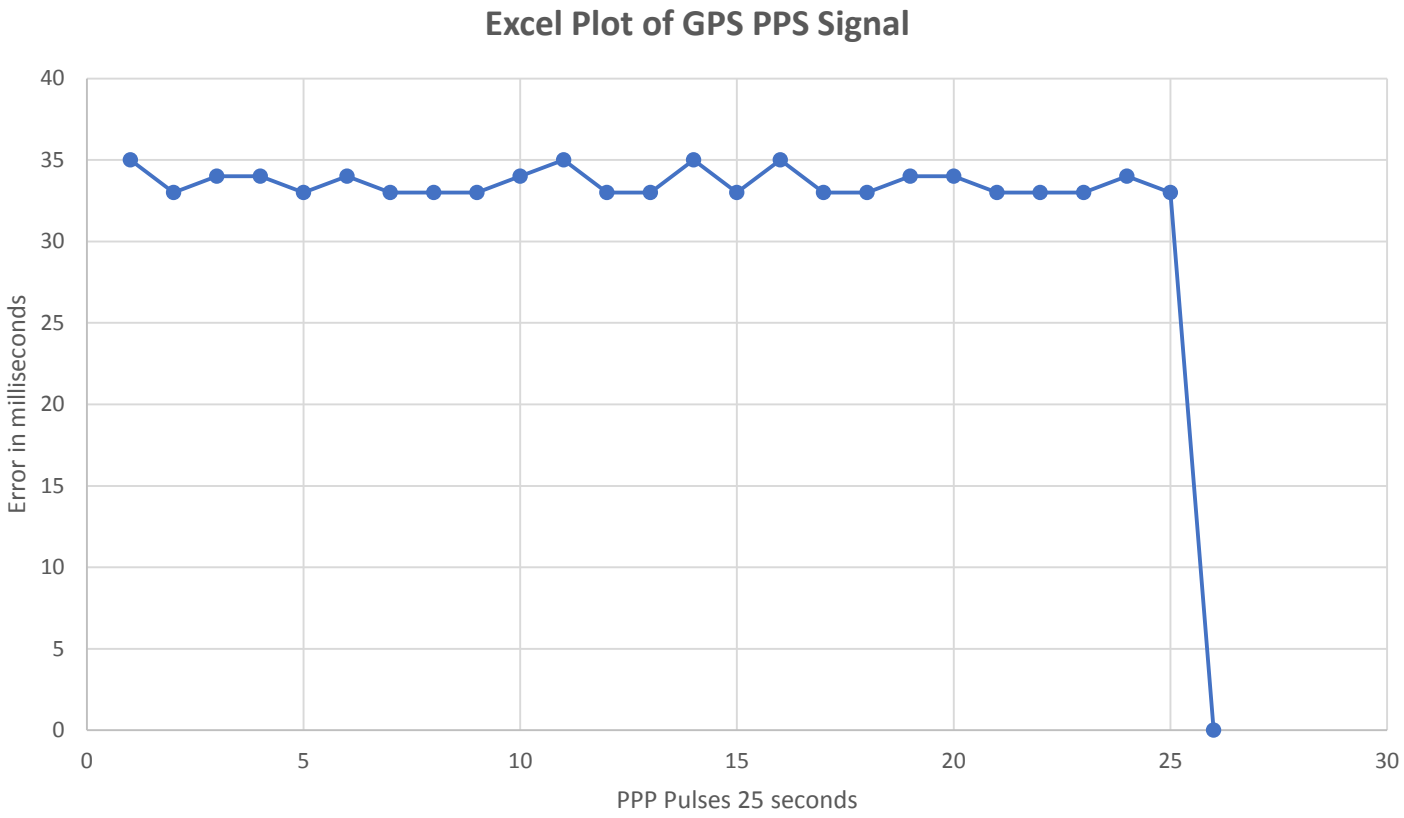
- Arduino Mega is connected to GPS receiver (placed on breadboard) like so:
  - Arduino 5V to GPS Vin
  - Arduino GND to GPS GND
  - Arduino PWM 2 to GPS PPS
- This setup ensures that data transmitted to the GPS is relayed to the Arduino Mega
- Objective is to capture an electrical signal that repeats once per second using the GPS module

Code used to record the GPS Receiver PPS Square Pulse



- Loop() function ensures that:
  - If digitalRead(2), or the value from pin 2, is a high value, the program prints “HIGH”
  - In contrast, if digitalRead(2) reads a low value, the program prints “LOW”
  - Continuous

PPS Signal Error



- Serial monitor data was collected for approx. 25 data points
- Data points were calculated using the following formula: 1 – [(pulse 2 start time) – (pulse 1 start time)]
- GPS PPS signal emits a consistent 33 to 35 millisecond error

References

- https://airu.coe.utah.edu/wp-content/uploads/sites/62/2017/09/adafruit-ultimate-gps.pdf
- https://www.tramsoft.ch/downloads/garmin/NMEA%20data.htm