

AeroQuad Flight Software Timing

November 2010

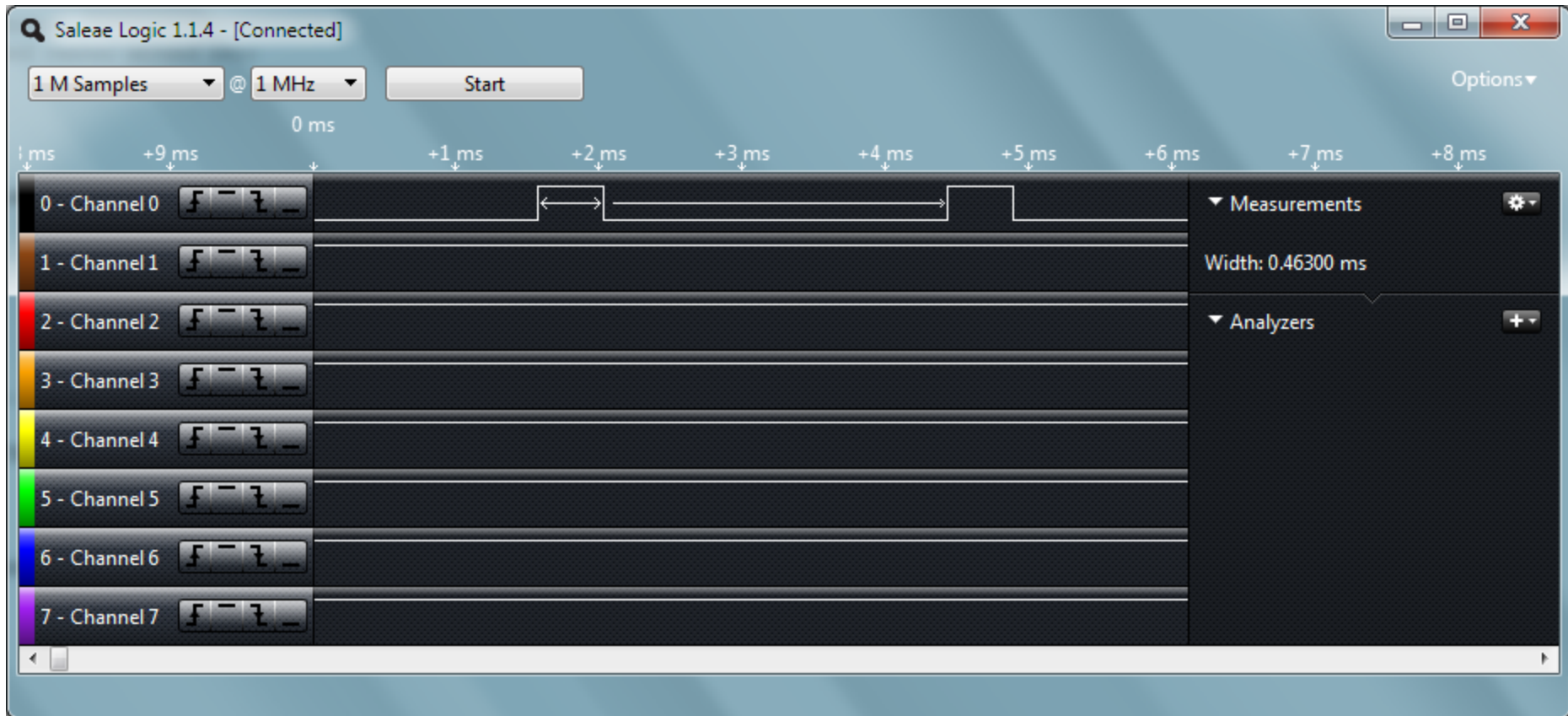
Outline

- Time each critical portion of the AeroQuad flight software loop
 - The timing of the new I2C based sensors will be documented
 - Gyro – ITG3200
 - Accel – BMA180
 - Magnetometer – HMC5843
 - Barometer – BMA085
- Test methodology
 - Turn on a digital pin before executing the function, then turn it off afterwards
 - Use a logic analyzer to measure the active high pulse

Timing Summary with I2C Sensors

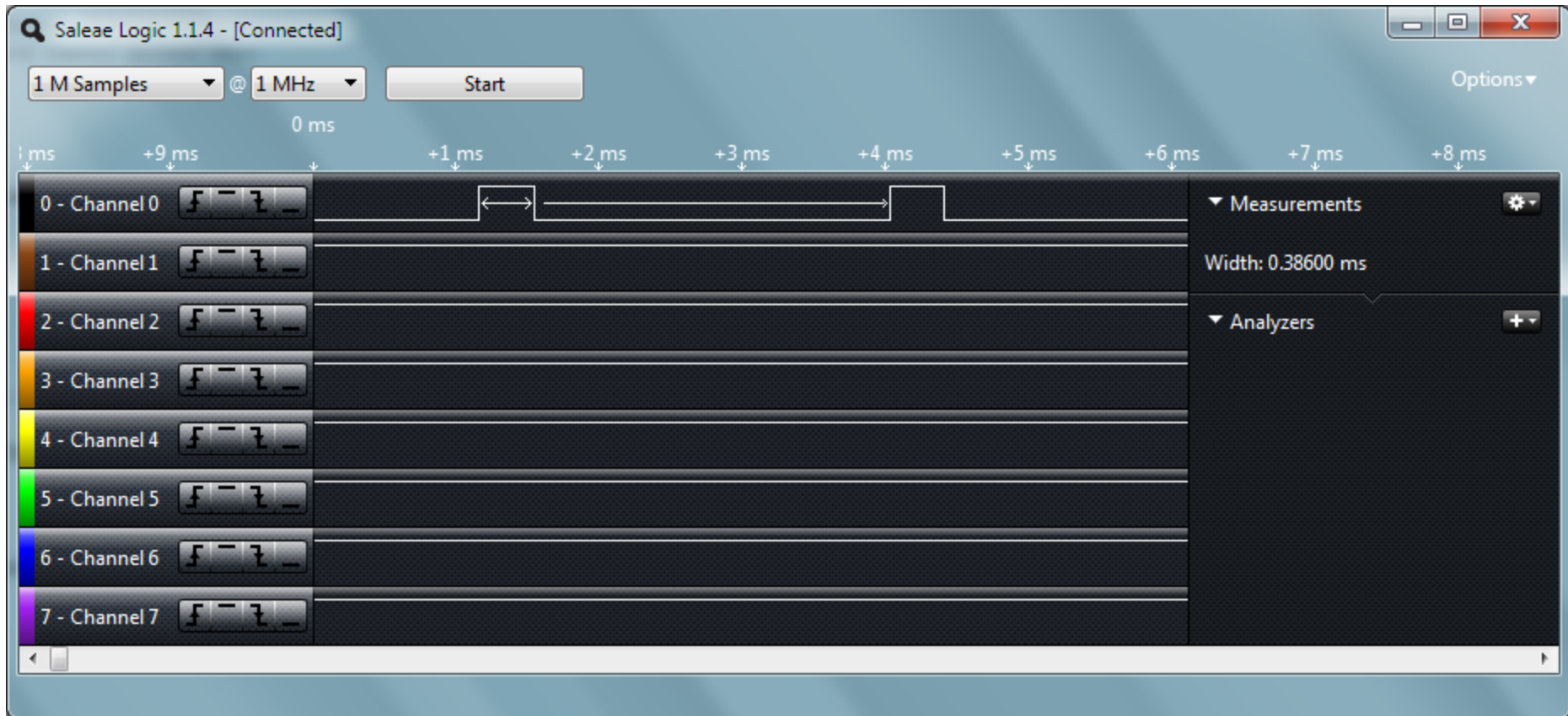
- Overall flight loop takes 3.756ms
 - Sensors = 2.85ms
 - Motor Commands = 0.503ms
 - Read Receiver = 0.403ms
- Takes 2.85ms to read flight sensors
 - Read accels and gyros only (0.849ms), then use DCM to calculate flight angle (2.001ms)
- Other sensors periodically affect flight loop timing
 - Barometer = 0.976ms every 26ms
 - Magnetometer = 1.289 every 125ms

Measure ITG3200 Gyros



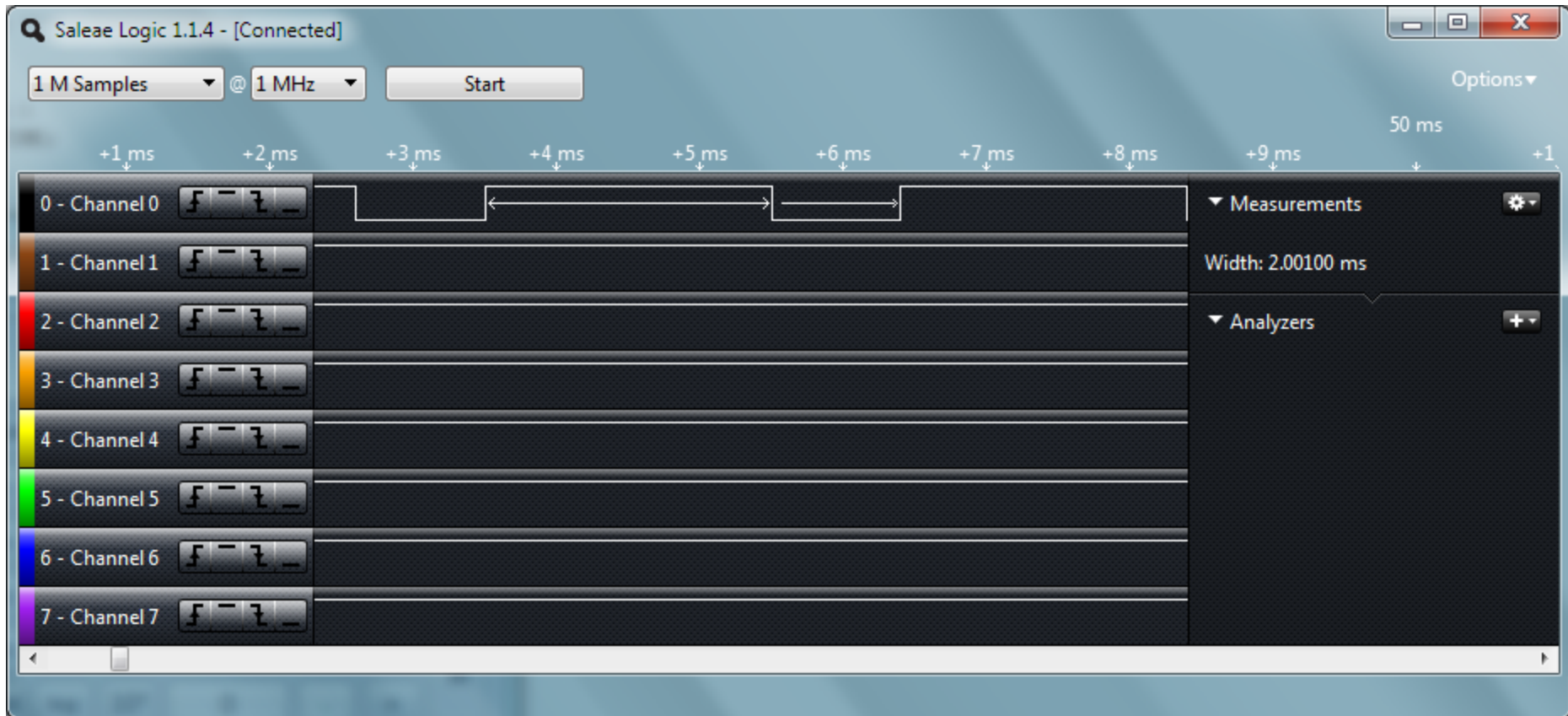
`gyro.measure() = 0.463ms`

Measure BMA180 Accelerometers



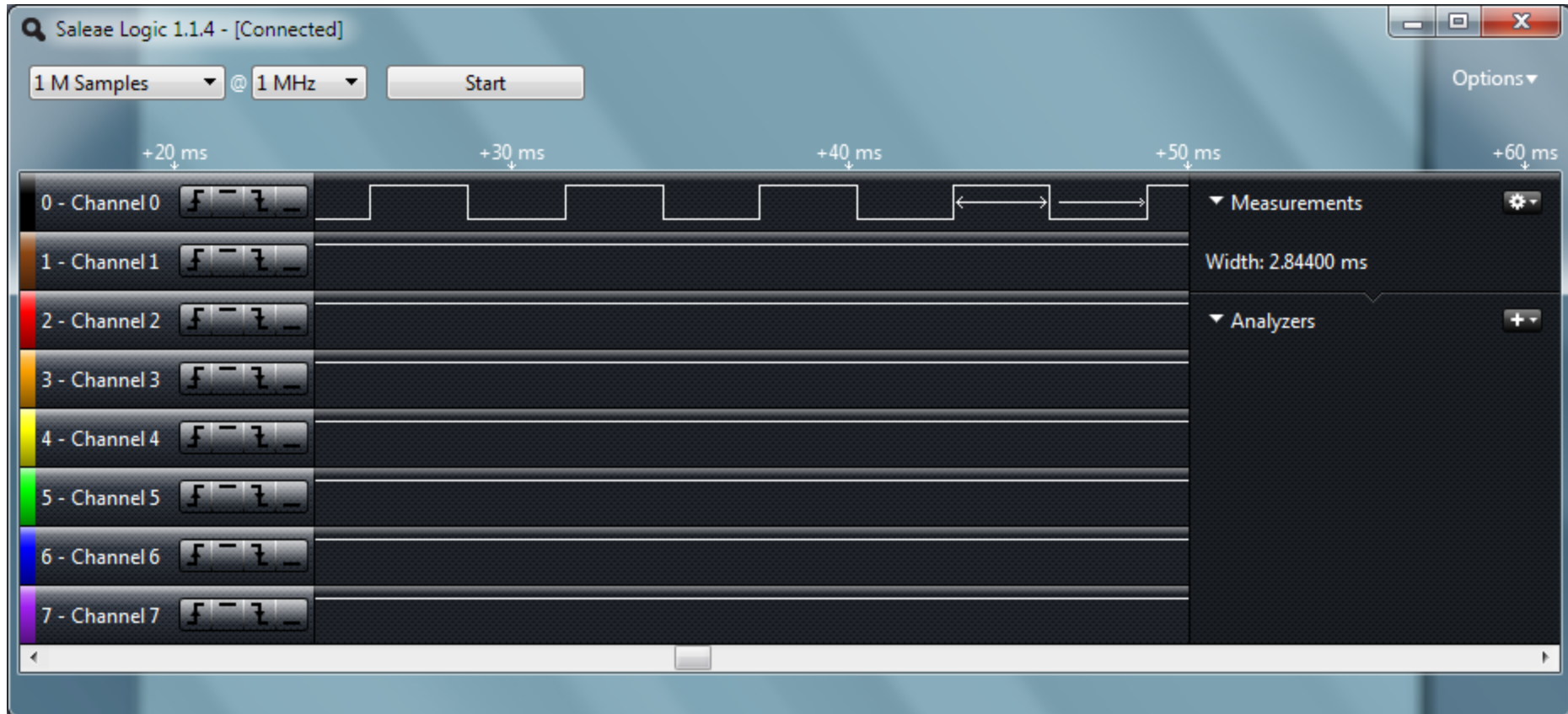
`accel.measure() = 0.386ms`

Duration of DCM Calculation



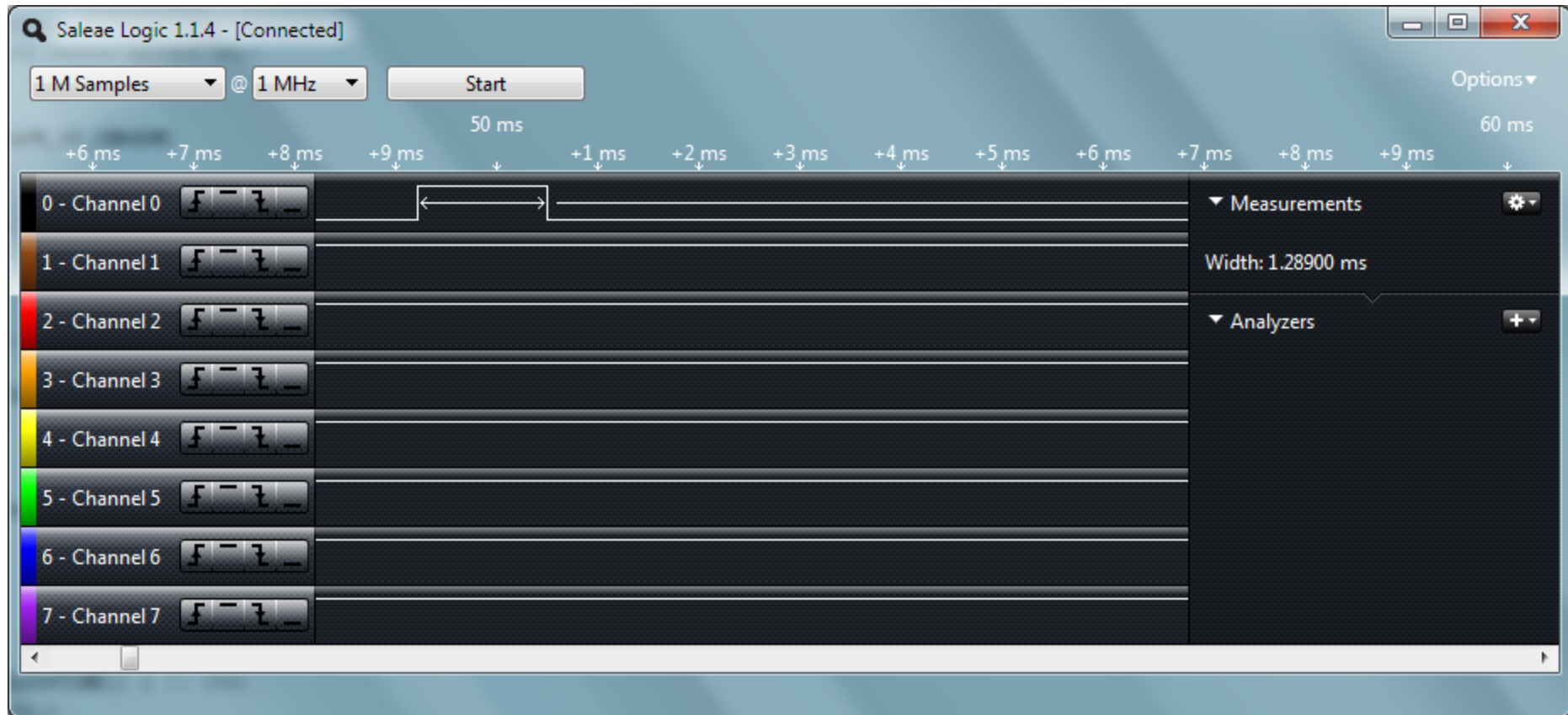
`flightAngle.calculate()` → DCM = 2.001ms

Timing Summary for Basic Sensors



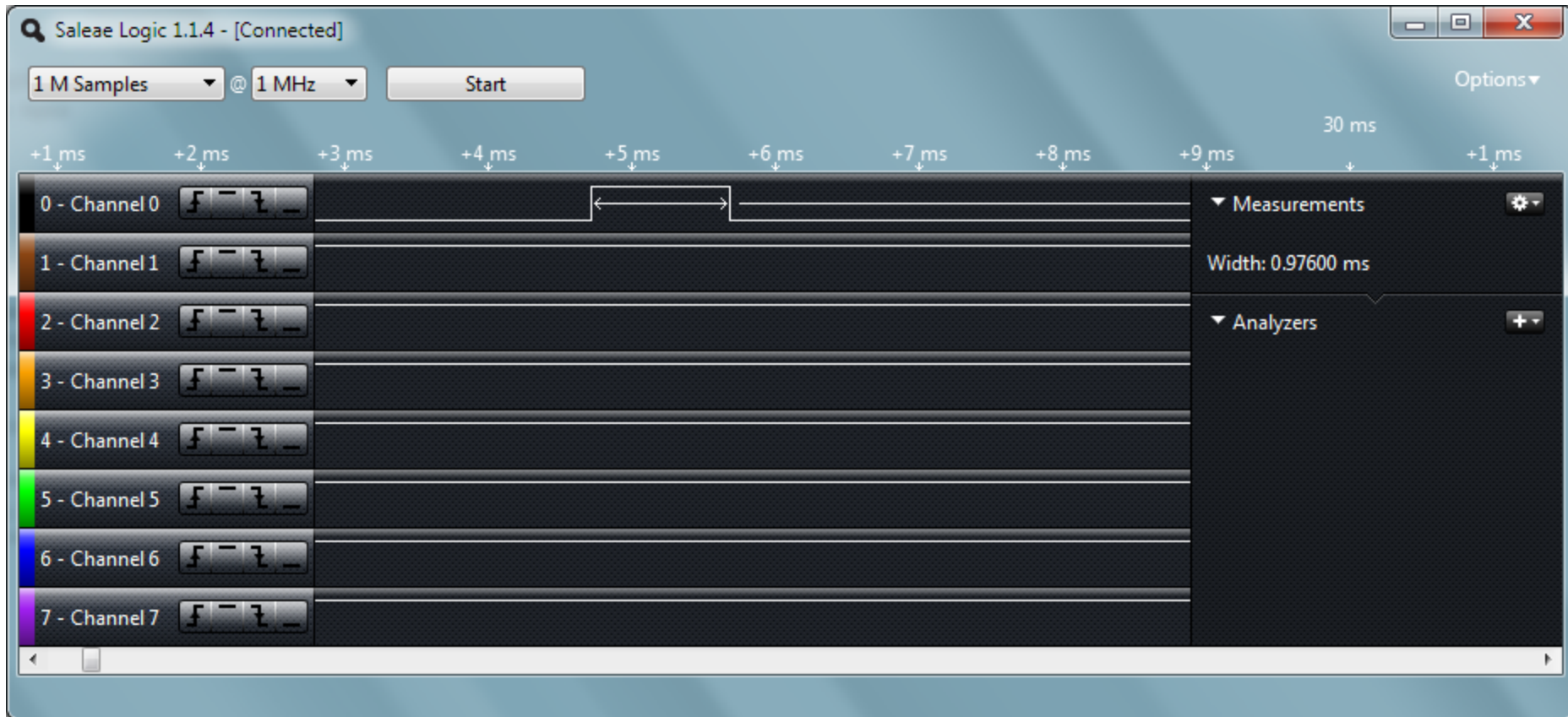
- `gyro.measure()` + `accel.measure()` + `flightAngle.calculate()` = 2.85ms
- Measured `readSensors()` with altitude and compass off = 2.844ms
- Individual measurements + overall function time duration agree.

Measure HMC5843 Magnetometer



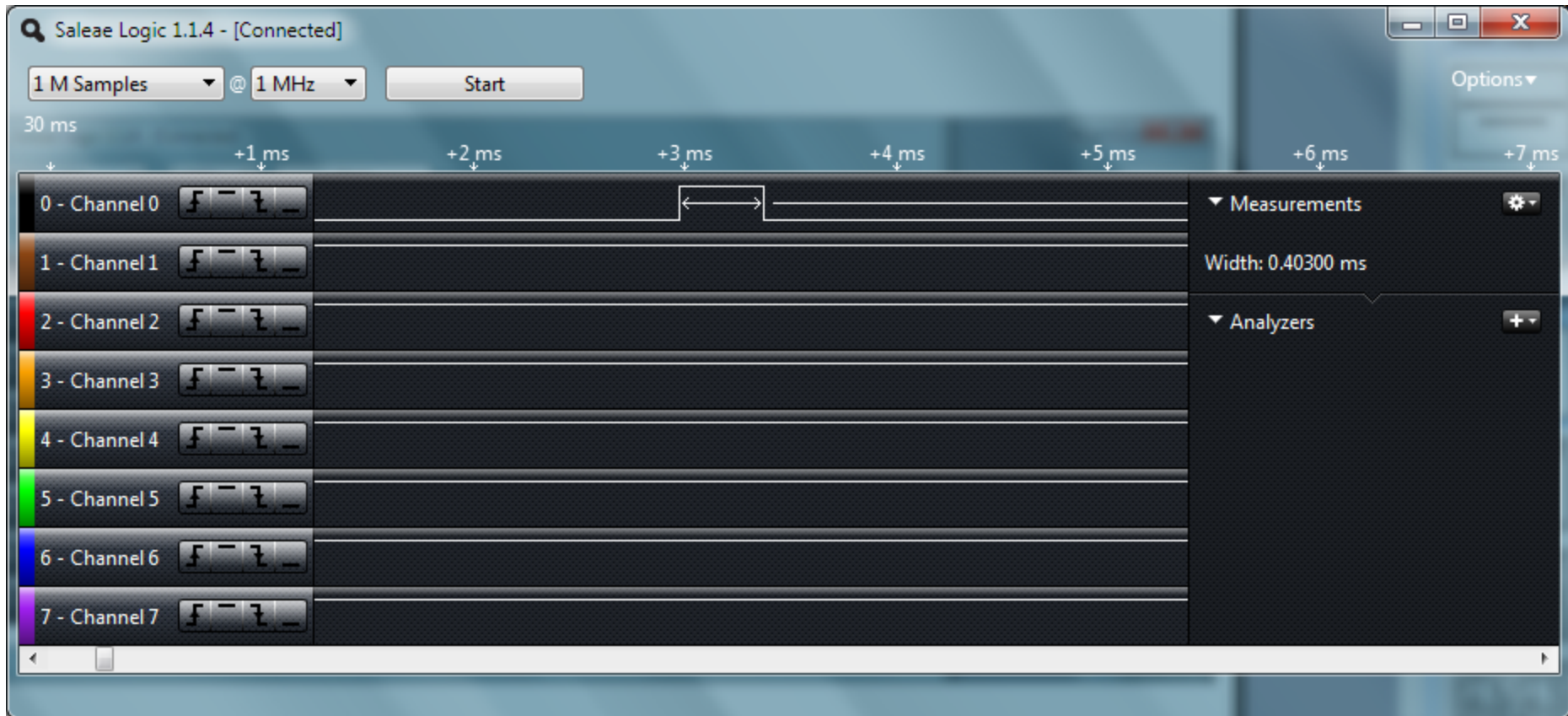
`compass.measure() = 1.289ms`

Measure BMA085 Barometer



`altitude.measure() = 0.976ms`

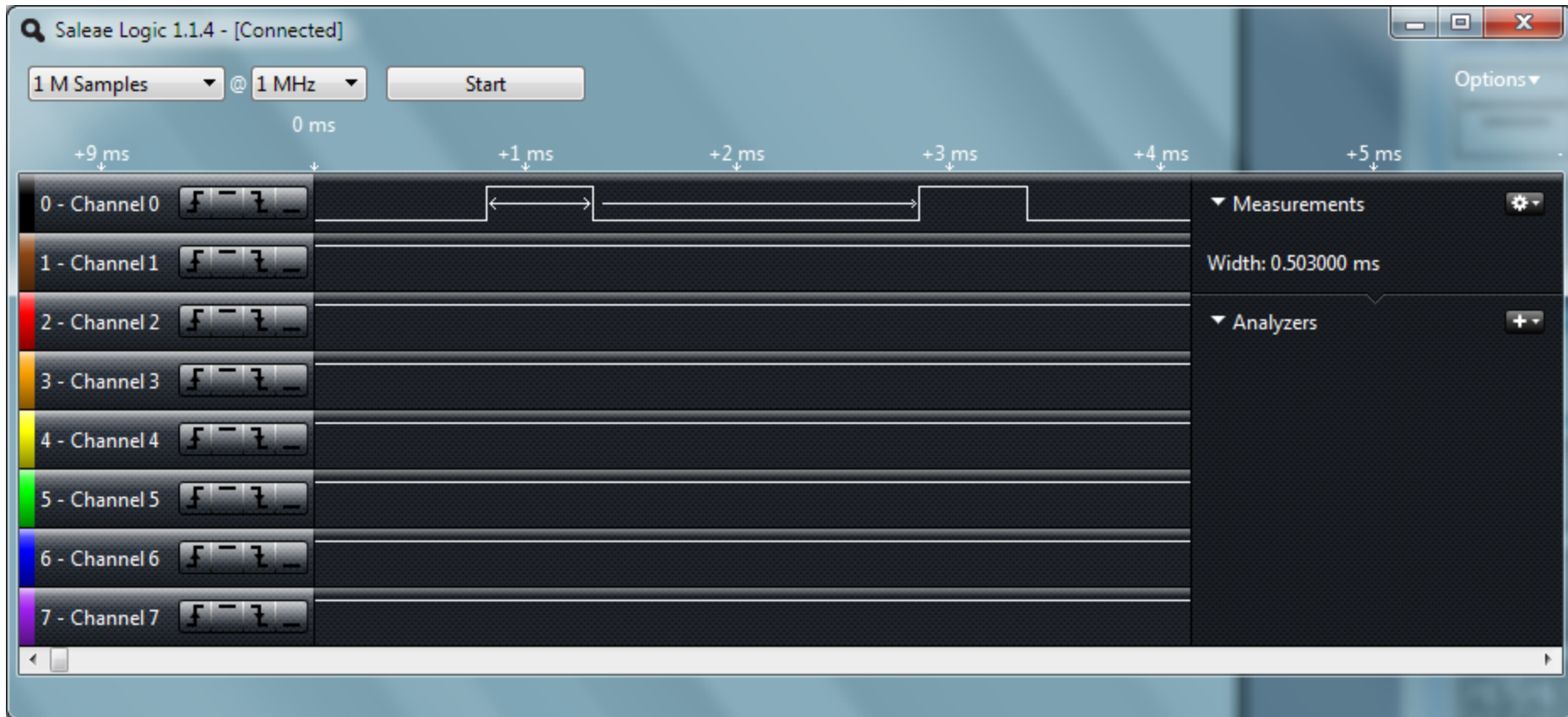
PCINT Read of Receiver Data



`ReadPilotCommands() = 0.403ms`

PCINT is actually used to measure pin value changes, the above function call Reads variables which store timing data as calculated during ISR

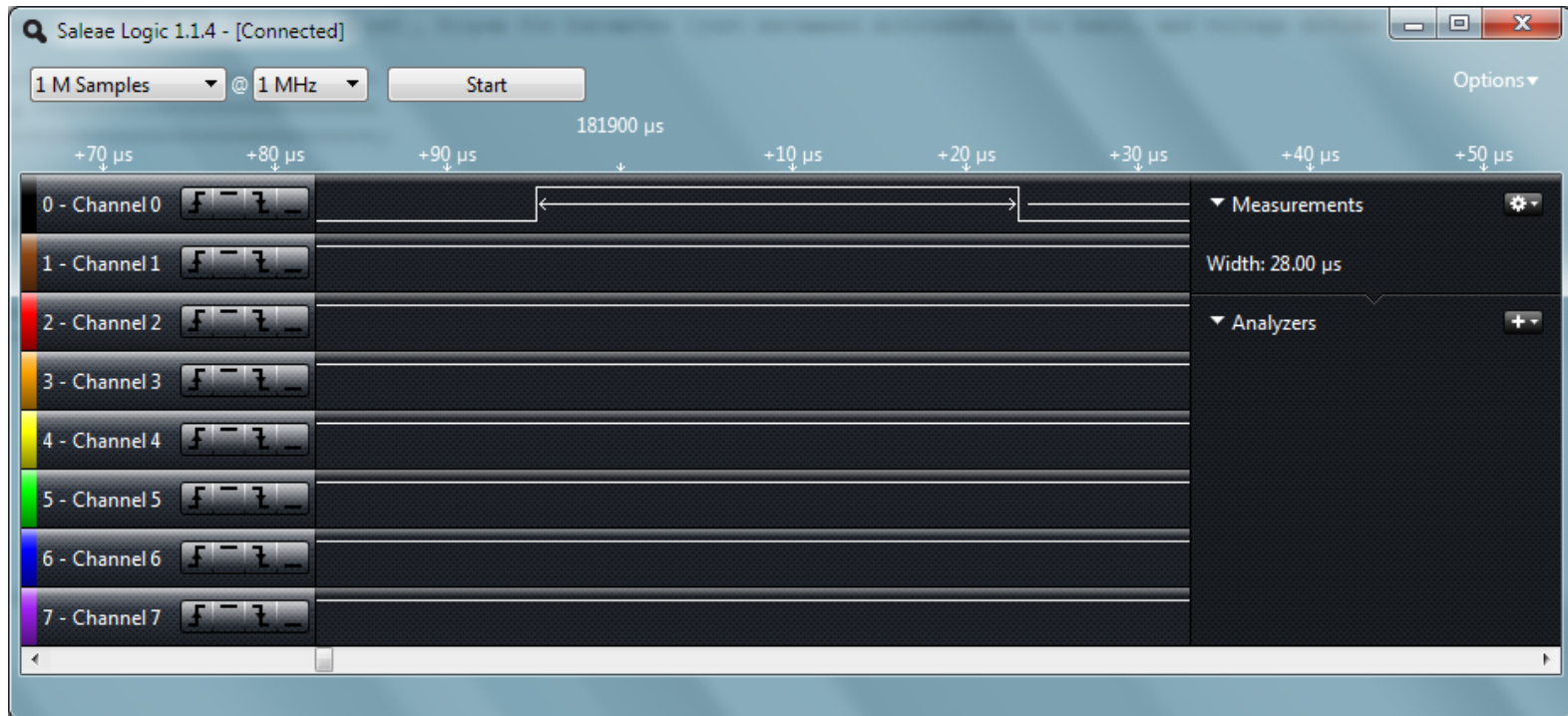
Duration of AnalogWrite() to Motors



`flightControl() = 0.503ms`

The above function uses `analogWrite()` to perform PWM update per channel (4)

Duration of Serial Command/Telemetry



wrapped with digital on/off
and measured with
logic analyzer

```
// Listen for configuration commands and reports telemetry
if ((telemetryLoop == ON) && (currentTime > telemetryTime)) {
    digitalWrite(LEDPIN, HIGH);
    readSerialCommand(); // defined in SerialCom.pde
    sendSerialTelemetry(); // defined in SerialCom.pde
    digitalWrite(LEDPIN, LOW);
    telemetryTime = currentTime + TELEMETRYLOOPTIME;
}
```

- Measured `readSerialCommand()` and `sendSerialTelemetry()`
- Duration = 28 μ s, implies that UART is loaded and hardware sends data but does not block flight loop