```
const int pinTRIG = 4;

unsigned long Hold_ON,  Hold_OFF;// us
unsigned int i, Ncycles;

char buffer[10];
char Ncycles_bytes[2], Hold_OFF_bytes[4], Hold_ON_bytes[4];

unsigned long tNow,  tCycleBegin;// us

void setup()
{
  pinMode(pinTRIG, OUTPUT);
  digitalWrite(pinTRIG, HIGH);

  // turn off pin13 LED
  pinMode(13, OUTPUT);
  digitalWrite(13, LOW);

  // initialise buffer
 memset(buffer, 0x00, sizeof(buffer));

  Serial.begin(115200, SERIAL_8N1);
  Serial.setTimeout(1000);
}


void loop()
{
  // nothing to do in here!
}


void MainCycle()
{
  // loop Ncycles doing trigger high low timing as requested
  for (i = 1; i <= Ncycles; i++)
  {
    // First hold OFF
    digitalWrite(pinTRIG, LOW);
     tCycleBegin =micros();
```

```
    while ((micros() - tCycleBegin) < Hold_OFF)
    {
      // do nothing until Hold_OFF time has elapsed
    }
    // Now hold ON
    digitalWrite(pinTRIG, HIGH);
     tCycleBegin =micros();
    while ((micros() - tCycleBegin) < Hold_ON)
    {
      // do nothing until Hold_ON time has elapsed
    }
    // inform PC of cycle number completed, in ASCII format
    Serial.println(i);
  }
    Serial.flush();
}


void serialEvent()
{
  if (Serial.peek() == '@') // beginning of instruction in correc
  {
    Serial.read(); // discard '@'
    // read all into buffer, then parse values as appropriate
    Serial.readBytesUntil('#', buffer, sizeof(buffer));// does N
    //Serial.write(reinterpret_cast<byte*>(&buffer), sizeof(buf

   Ncycles_bytes[0] = buffer[1];
   Ncycles_bytes[1] = buffer[0];
   memcpy(&Ncycles, &Ncycles_bytes, sizeof(Ncycles));

   Hold_OFF_bytes[0] = buffer[5];
   Hold_OFF_bytes[1] = buffer[4];
   Hold_OFF_bytes[2] = buffer[3];
   Hold_OFF_bytes[3] = buffer[2];
   memcpy(&Hold_OFF, &Hold_OFF_bytes, sizeof(Hold_OFF));

   Hold_ON_bytes[0] = buffer[9];
   Hold_ON_bytes[1] = buffer[8];
   Hold_ON_bytes[2] = buffer[7];
   Hold_ON_bytes[3] = buffer[6];
```

```
    memcpy(&Hold_ON, &Hold_ON_bytes, sizeof(Hold_ON));

     MainCycle();
   }
   else
   {
     // discard entire buffer
     while (Serial.available() > 0)
     {
     Serial.read();
     }
   }
}
```