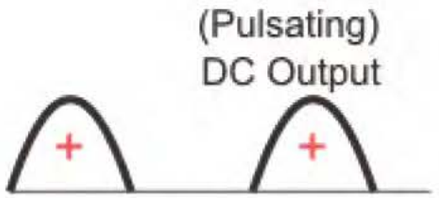
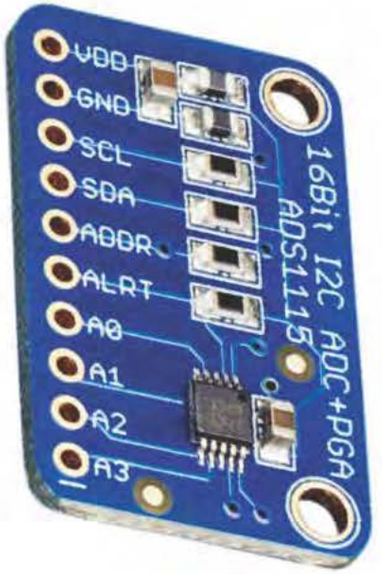
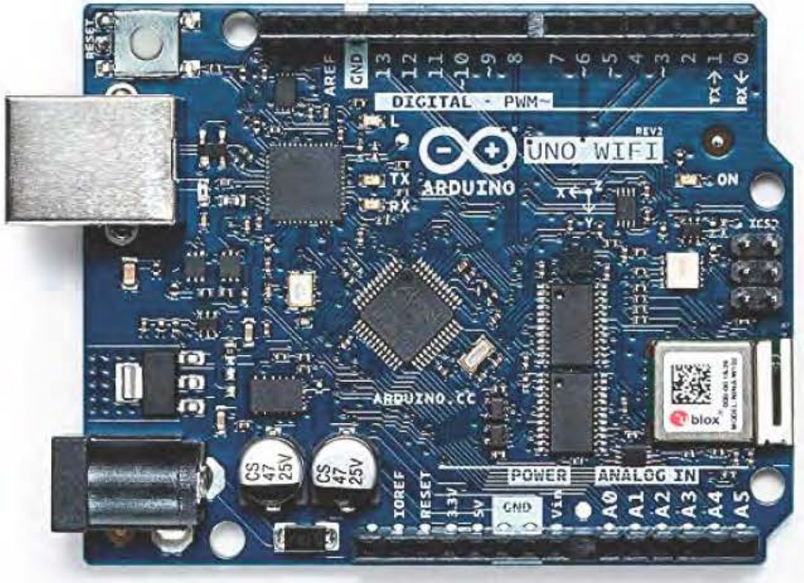
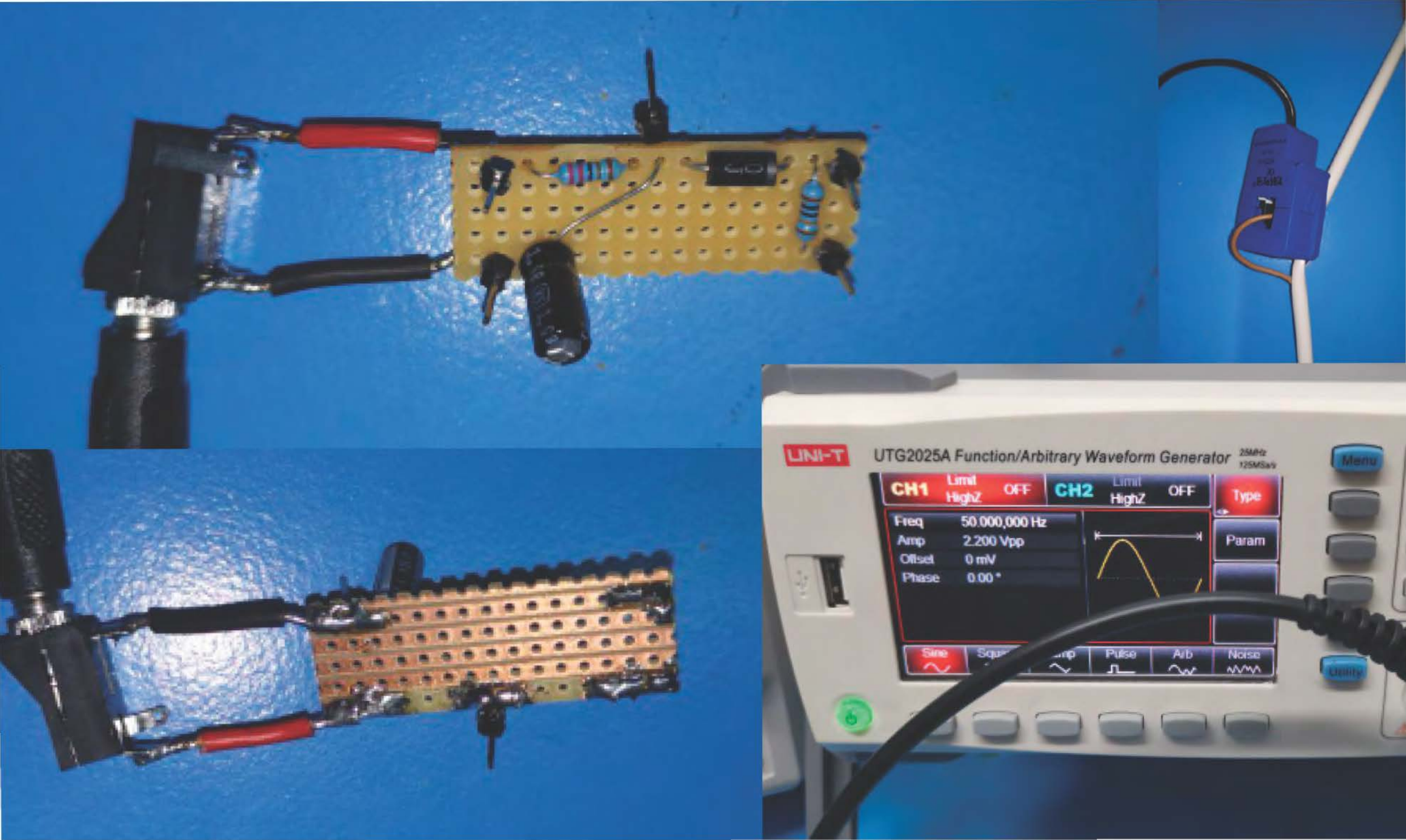
Hardware

- SCT013-005 split core current transformer 5A/1V, built-in resistance
- RC low-pass filter : capacitor 1uF + resistor 2.2kΩ (filter from 70Hz) -> noise management
- Half wave rectifier : Schottky diode MBR2200 DO-15 200 V + resistor 100kΩ
- ADC Arduino ADS1115/1015
- Arduino UNO Wifi Rev2
- female audio jack 3.5 mm








UNI-T
UTG2025A Function/Arbitrary Waveform Generator
CH1 Limit HighZ OFF
CH2 Limit HighZ OFF
Freq 50.000,000 Hz
Amp 2.200 Vpp
Offset 0 mV
Phase 0.00 °
Type
Param
Menu
Utility
Sine
Pulse
Arb
Noise

Signal from function generator

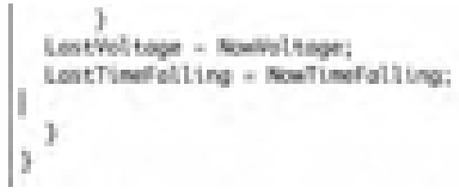
Signal from SCT-013

Code: From Peak Detection to Frequency

setup based on ADS_continuous.ino for the ADS1X15 Library

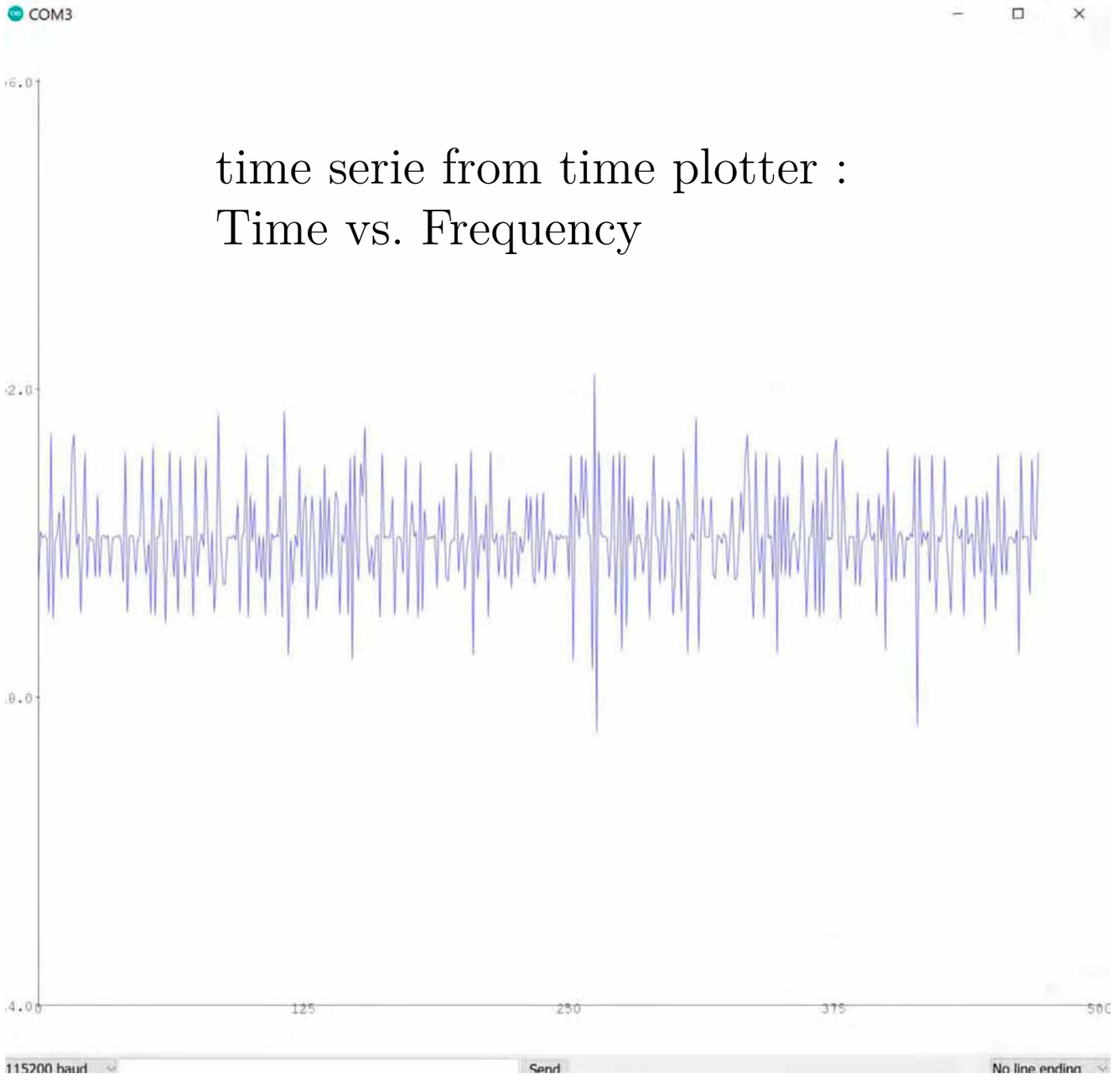
```
// Split Core SCT013 5A/1V an L1 anschliessen
// L und K an LowPassFilter 70 Hz
// Halbwellengleichrichter (Half Bridge Rectifier)
// L an A0 des Ads1015 K an GND des ADS1015
// ADS1015 an Arduino anschliessen (siehe Adafruit Webseite)

#include "ADS1X15.h"
// choose you sensor
// ADS1013 ADS(0x48);
// ADS1014 ADS(0x48);
 ADS1015 ADS(0x48);
// ADS1113 ADS(0x48);
// ADS1114 ADS(0x48);
//ADS1115 ADS(0x48);

// Definitionen
int32_t baudrate = 115200;
const float multiplier = 0.0625;// 0.015625 8// 0.03125 4 //0.0625 2 // Gain Factor (4our) // 0.1132changer????
int32_t LastVoltage = 0;   // Integer Werte !! müssen konvertiert werden zu Voltage
int32_t NowVoltage = 0;   // Integer Werte !! müssen konvertiert werden zu Voltage
int32_t Threshold = 500;
bool Falling = false, Rising = false, Falling_Edge, Rising_Edge;
long TimeMS = 0, NowTimeFalling = 0, NowTimeRising = 0, LastTimeFalling = 0, LastTimeRising = 0, PeriodFalling = 0, PeriodRising =0;
float FrequencyFalling = 0, FrequencyRising = 0, Period;

void setup()
{
  Serial.begin(baudrate);
  ADS.begin();
  delay(1000);
  ADS.setGain(8);      // Gain eight
  ADS.setDataRate(7);  // fast
  ADS.setMode(0);      // continuous mode
  ADS.readADC(0);      // first read to trigger
  delay(1000);
  Wire.setClock(400000);
  delay(1000);

}
```

```
void loop()
{
  NowVoltage = ADS.getLastValue(); // reading voltage as Bits
if (NowVoltage > Threshold){
    if (NowVoltage < LastVoltage){
        // Serial.print(" Falling");
        if (Falling == false){
          Falling_Edge = true;
          Rising_Edge = false;
          // Falling = true;
          // Serial.print(" FALLING EDGE DETECTED! ");
          NowTimeFalling = micros();
          TimeMS = millis();
          PeriodFalling = NowTimeFalling - LastTimeFalling;
          Period = PeriodFalling; // Konvertierung Int to Float
          FrequencyFalling = 1000000 / Period;
          if ((FrequencyFalling < 60) && (FrequencyFalling > 40)) {
          // Serial.print(" FREQUENCY_FALLING: ");
          Serial.print(TimeMS);
          Serial.print(",");
          Serial.print(NowTimeFalling);
          Serial.print(",");
          Serial.println(FrequencyFalling);
          // Serial.println("/n");

          }
        }
        Falling = true;
        Rising = false;
      }
      else if (NowVoltage > LastVoltage) {
        // Serial.print(" Rising");
        if (Rising == false){
          Rising_Edge = true;
          Falling_Edge = false;
          // Rising = true;
          // Serial.print(" RISING EDGE DETECTED! ");
        }
        Rising = true;
        Falling = false;
```

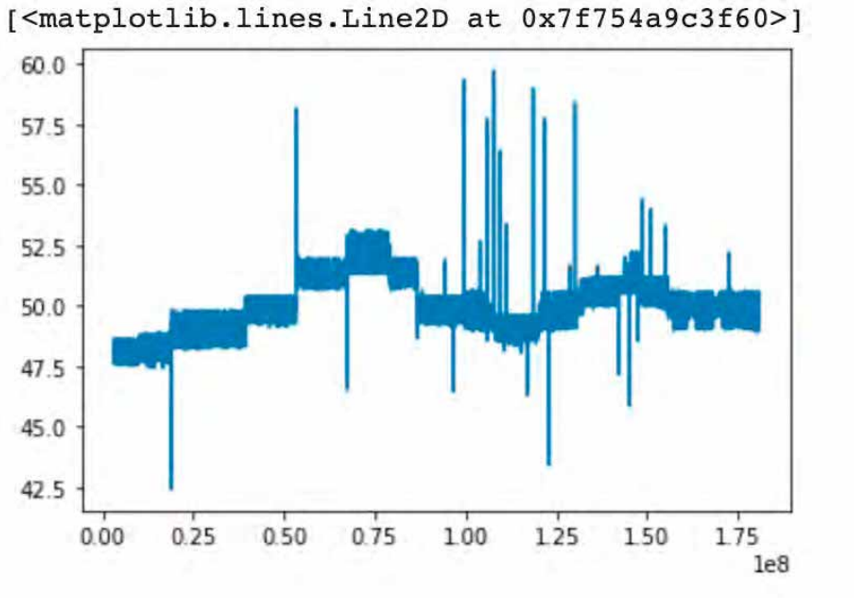


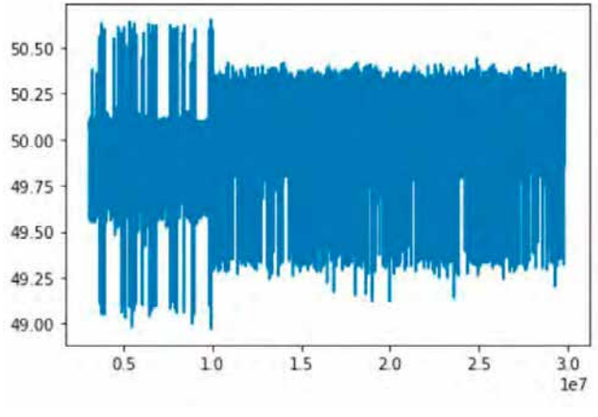
possible to measure frequency at rising point

Plot



1. Variation of frequency from 48 to 52hz with function generator

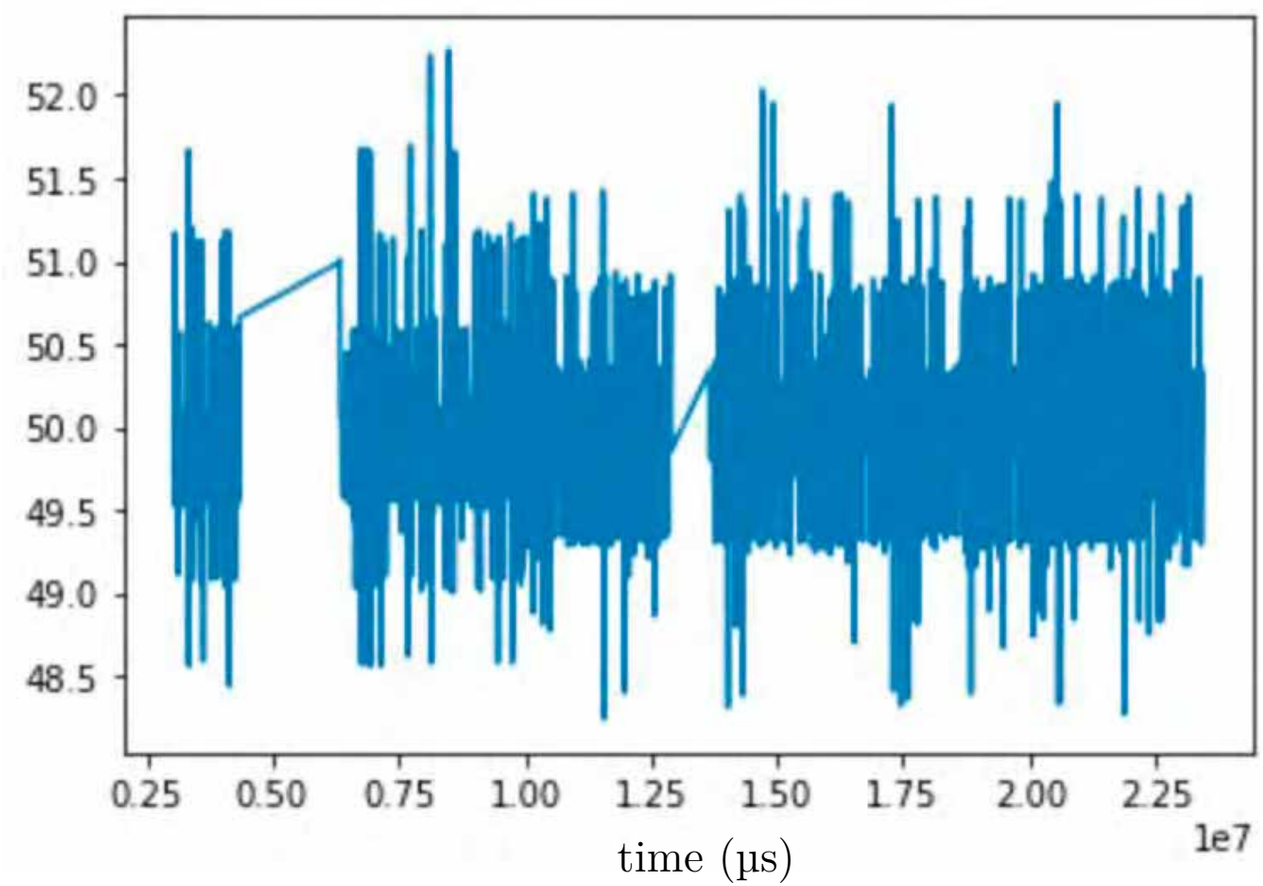




time serie function gen. at 50Hz

Time (µs) vs. Frequency (Hz)
of radiator

Frequency
Hz



Concerns & Improvements

- Not real-time, skipping measurements! Noisy (where slope=0)!-> Oscilloscope + save data, other plug-in sensors
- SCT-013 CTs : voltage low -> less precision, noisy
- Other ADC (measurement in differential mode)?
- Minimise connections, shorter
- RTC clock/masterclock?
- Half wave rectifier — voltage drop after diode + signal distorted at bottom.
- Code— initialization function, interrupts: measuring at equally spaced intervals, overflow -> while statement
- Calibration with function generator -> moving average? sampling rate injects noise in time serie
- Further reading: FFT, sampling rates, resolution

Alternative circuits

- OpAmplifier as voltage follower + DC offset (2 resistors + capacitor -> midpoint between GND and Vcc.
- Voltage transformers: Single Phase AC Voltage Module, ACS712 Current Sensor Module, Voltage-to-frequency converters + optocouplers