// PROJECT  :WireLevelSQWPlotter

// PURPOSE  :Plotting the Square Wave output from the DS1307RTC

// AUTHOR   :C. D'Arcy

// DATE     :Created 2020 12 10. Confirmed 2025 03 02.

// uC       :\*

// STATUS   :Working

// REFERENCE:http://darcy.rsgc.on.ca/ACES/Datasheets/DS1307.pdf

#include <Wire.h>

#define RTCADDRESS  0x68

#define SQWREGISTER 7

#define SQWE        4

#define RS0         0

#define RS1         1

//correct way to assemble bit flags

#define Hz1   0 << RS1 | 0 << RS0   // 1 Hz (2^0)

#define Hz4k  0 << RS1 | 1 << RS0   // 4.096 kHz (2^12)

#define Hz8k  1 << RS1 | 0 << RS0   // 8.192 kHz (2^13)

#define Hz32k 1 << RS1 | 1 << RS0   // 32.768 kHz (2^15)

#define INTERRUPTPIN 2

volatile uint8\_t count = 0;

void setup() {

  Serial.begin(1200);  //reduce to 1200 baud for viewability....

  while (!Serial);

  Wire.begin();

  Serial.println(setSQW(Hz1));

  pinMode(INTERRUPTPIN, INPUT\_PULLUP); // no need for pullup resistor

  attachInterrupt(digitalPinToInterrupt(INTERRUPTPIN), ISR\_SQW, FALLING);

}

uint8\_t setSQW(uint8\_t frequency) {

  Wire.beginTransmission(RTCADDRESS);

  Wire.write(SQWREGISTER);

  Wire.write(1 << SQWE | frequency);  // RS1:RS0=00 selects 1Hz

  Wire.endTransmission();

  // optional: read it back for verification/confirmation...

  Wire.beginTransmission(RTCADDRESS);

  Wire.write(SQWREGISTER);

  Wire.endTransmission();

  Wire.requestFrom(RTCADDRESS, 1);

  while (!Wire.available());

  uint8\_t value = Wire.read();

  return value;

}

void ISR\_SQW() { count++;}

void loop() {

  Serial.print("DS1307\_SQW:");    //adds a label to the Plotter

  Serial.println(count % 2);      //0 or 1

}

