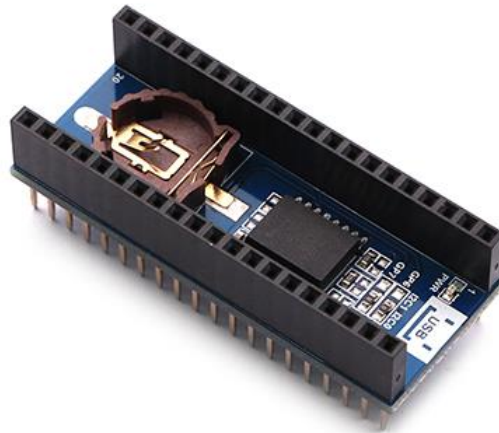


RaspberryPi

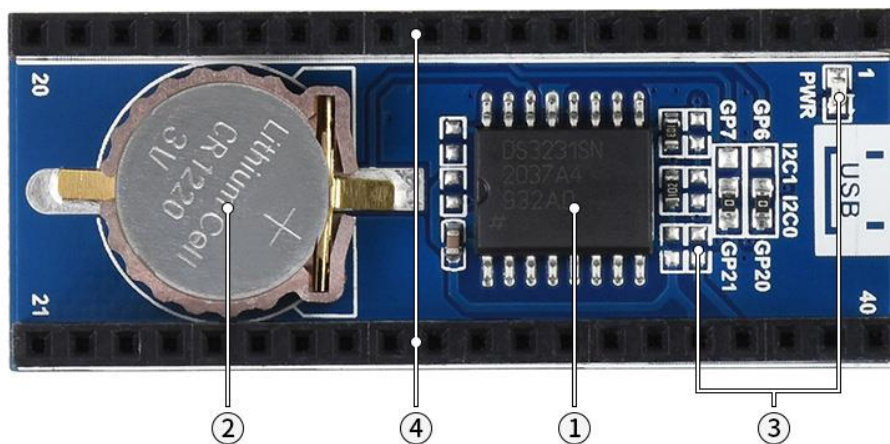
Precision RTC Module for Pico

Incorporates High Precision

User Manual



What's on Board:



1- DS3231

high precision RTC chip, I2C bus

2- RTC backup battery

supports CR1220 button cell

3- Power indicator

- enabled by soldering a 0Ω resistor on the jumper, disabled by default

4- Raspberry Pi Pico header

for attaching to Raspberry Pi Pico, stackable design

Pinout Definition:

GP0	1	40	VBUS
GP1	2	39	VSYS
GND	3	38	GND
GP2	4	37	3V3_EN
GP3	5	36	3V3(OUT)
GP4	6	35	ADC_VREF
GP5	7	34	GP28
GND	8	33	GND
GP6	9	32	GP27
GP7	10	31	GP26
GP8	11	30	RUN
GP9	12	29	GP22
GND	13	28	GND
GP10	14	27	GP21
GP11	15	26	GP20
GP12	16	25	GP19
GP13	17	24	GP18
GND	18	23	GND
GP14	19	22	GP17
GP15	20	21	GP16



3V3(OUT)	Power positive
GND	Ground
GP20	SDA I2C SDA pin
GP21	SCL I2C SCL pin

Raspberry Pi Code:

1. Open a terminal of Raspberry Pi
2. Download and unzip the demo codes to directory Pico C/C++ SDK

#Note that the directory of SDK may be different for different users, you need to check the actual directory. Generally, it should be ~/pico/.

```
wget -P ~/pico https://www.waveshare.com/w/upload/2/26/Pico-rtc-ds3231_code.zip
```

```
cd ~/pico
```

```
unzip Pico-rtc-ds3231_code.zip
```

1. Hold the BOOTSEL button of Pico, and connect the USB interface of Pico to Raspberry Pi then release the button.
2. Compile and run the pico-rtc-ds3231 examples

```
cd ~/pico/pico-rtc-ds3231_code/c/build/
```

```
cmake ..
```

```
make
```

```
sudo mount /dev/sda1 /mnt/pico && sudo cp rtc.uf2 /mnt/pico/ && sudo sync && sudo umount /mnt/pico && sleep 2 && sudo minicom -b 115200 -o -D /dev/ttyACM0
```

3. Open a terminal and user minicom to check the sensor's information.

```
Welcome to minicom 2.7.1

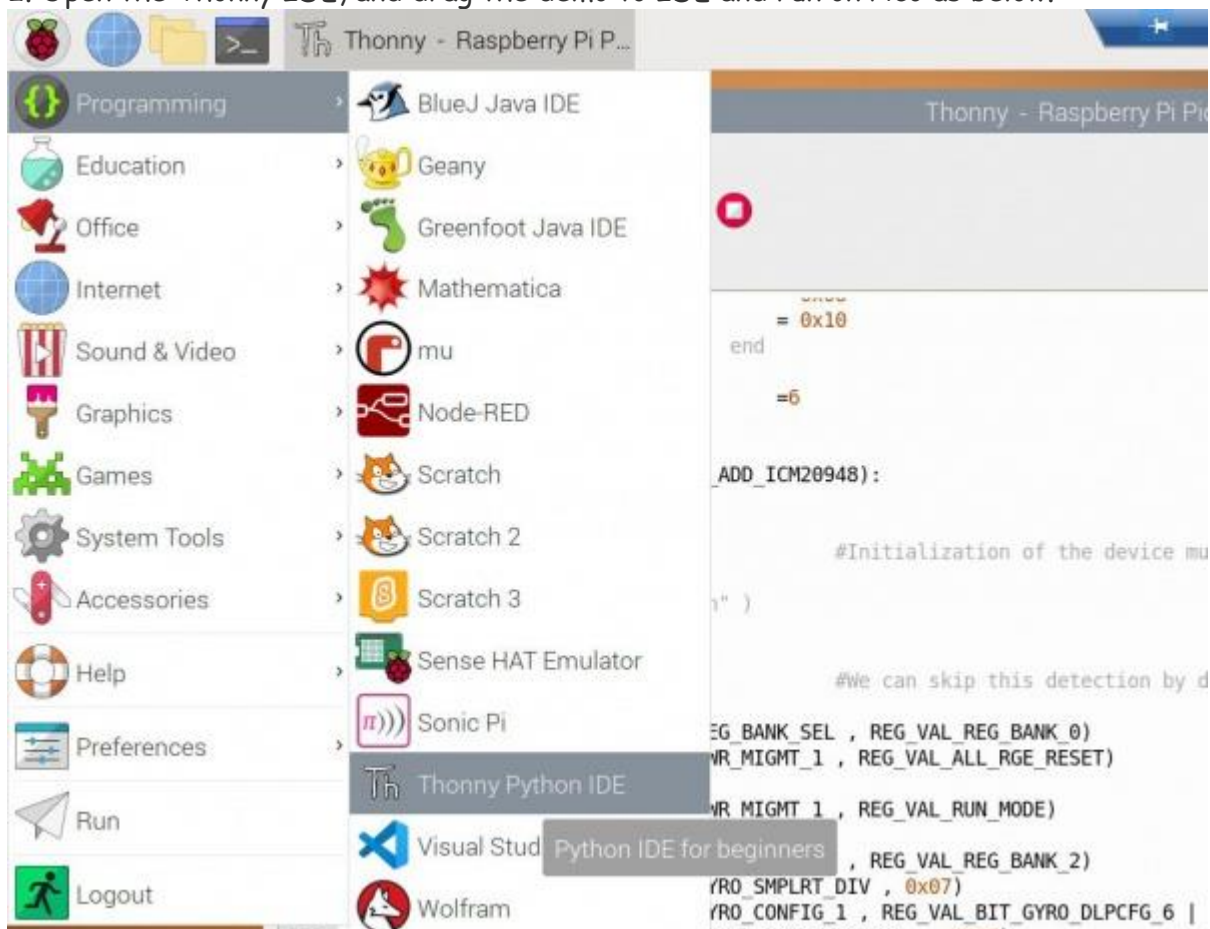
OPTIONS: I18n
Compiled on Aug 13 2017, 15:25:34.
Port /dev/ttyACM0, 21:34:41

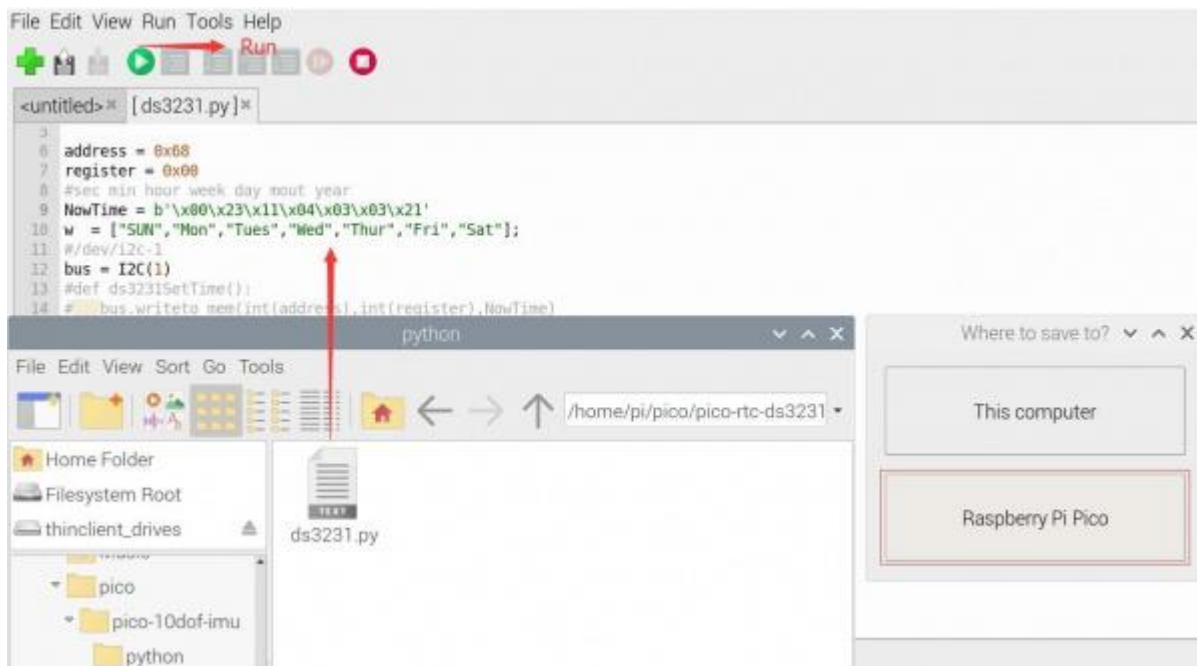
Press CTRL-A Z for help on special keys

2021/02/05 13:49:24 Fri
2021/02/05 13:49:25 Fri
2021/02/05 13:49:26 Fri
2021/02/05 13:49:27 Fri
2021/02/05 13:49:28 Fri
2021/02/05 13:49:29 Fri
█
```

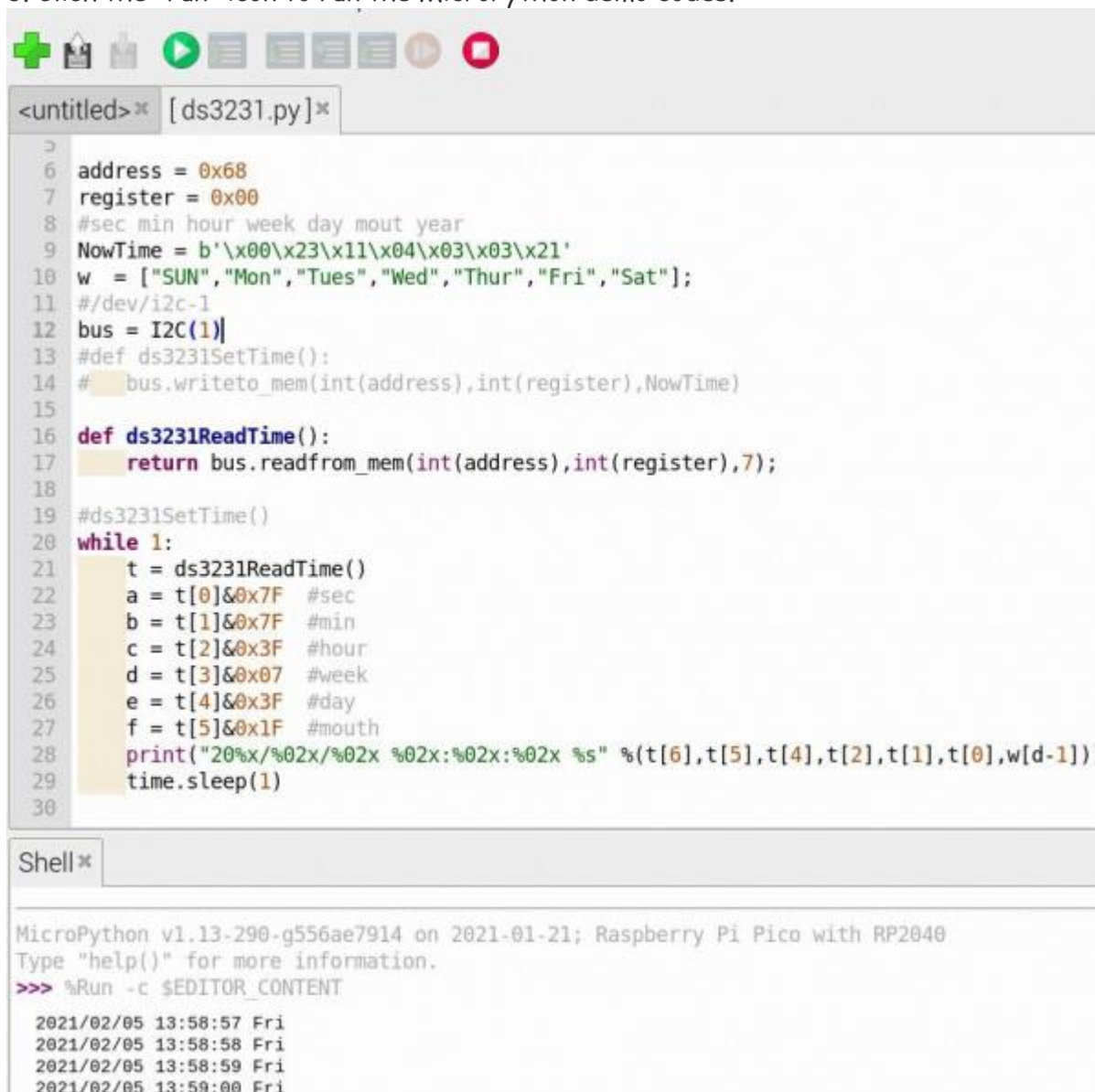
Python:

1. Refer to Raspberry Pi's guides to setup Micropython firmware for Pico
2. Open the Thonny IDE, and drag the demo to IDE and run on Pico as below.



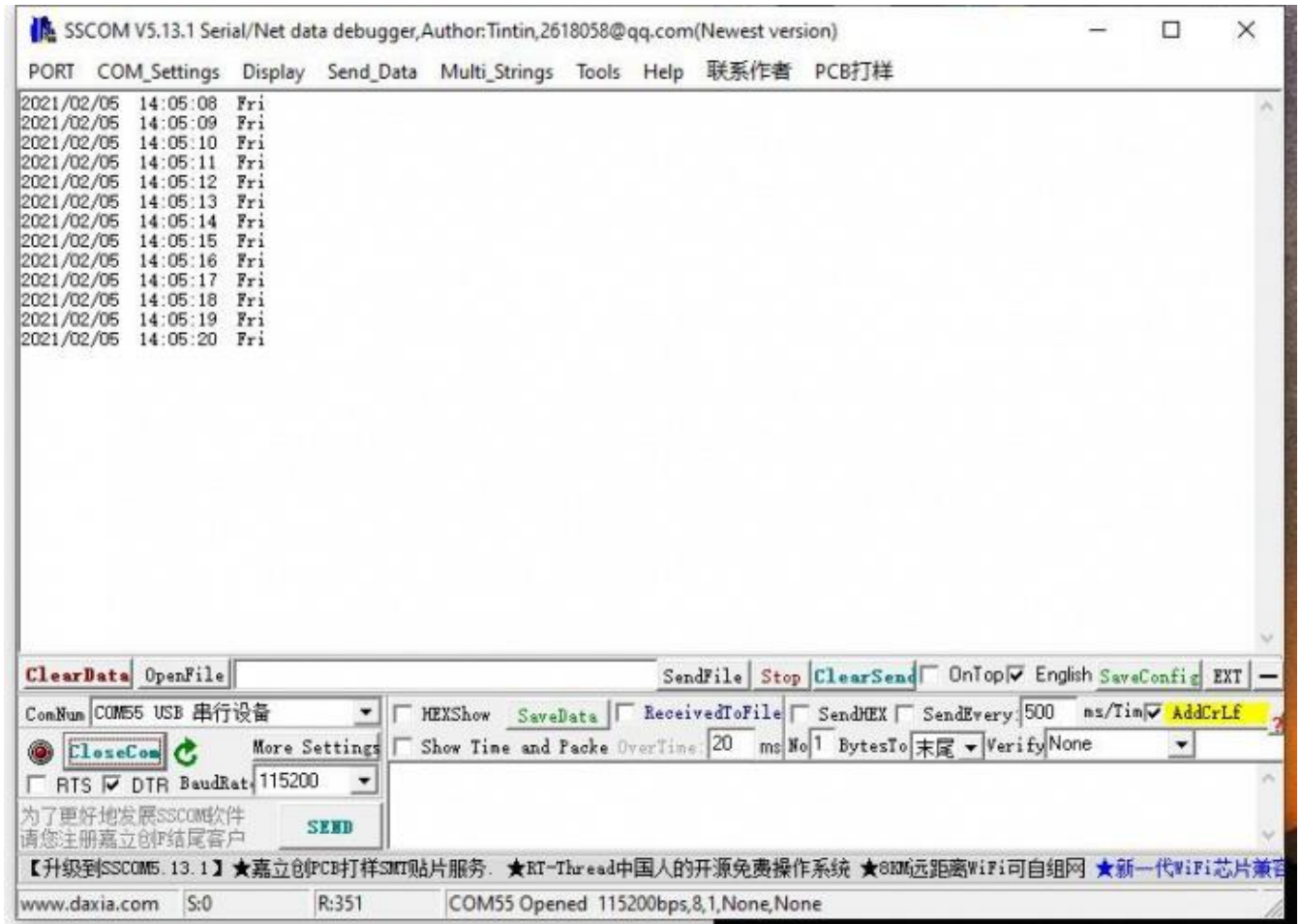


3. Click the "run" icon to run the MicroPython demo codes.



Windows:

- Download and unzip the [demo](#) to your Windows desktop, refer to Raspberry Pi's guides to set up the Windows software environment settings.
- Press and hold the BOOTSEL button of Pico, connect the USB of Pico to the PC with a MicroUSB cable. Import c or python program into Pico to make it run.
- Use the serial tool to view the virtual serial port of Pico's USB enumeration to check the print information, the DTR needs to be opened, the baud rate is 115200, as shown in the picture below:



Others:

- The LED light are not used by default, if you need to use it, you can solder a OR resistor on the R8 position. Click to view the [schematic diagram](#).
- The INT pin of DS3231 is not used by default. if you need to use it, you can solder the OR resistor on the R5,R6,R7 positions. Click to view the [schematic diagram](#).
 - Solder the R5 resistor, connect the INT pin to the GP3 pin of Pico, to detect the output status of the DS3231 alarm clock.
 - Solder the R6 resistor, connect the INT pin to the 3V3_EN pin of Pico, to turn off the Pico power when the DS3231 alarm clock outputs low level.
 - Solder the R7 resistor, connect the INT pin to the RUN pin of the Pico, to reset Pico when DS3231 alarm clock outputs low level.

This diagram illustrates the hardware setup for a Pico-RTC-DS3231 module, showing the connection of the DS3231 RTC to a Raspberry Pi Pico microcontroller.

Component Identification

H2 Identification

I2C0	GP20	H3	H1	GP21
	P SDA	1	1	P SCL
I2C1	GP6	2	2	GP7
		3	3	

Pin Connections

U1: Pi pico (40P)

Pin	GPIO	Function	Pin	GPIO	Function
GP0	1	GPIO0	VBUS	40	VBUS
GP1	2	GPIO1	VSYS	39	VSYS
GND	3	GND	GND	38	GND
GP2	4	GPIO2	3V3_EN	37	3V3_EN
GP3	5	GPIO3	3V3	36	3V3(OUT)
GP4	6	GPIO4	ADC_VREF	35	ADC_VREF
GP5	7	GPIO5	IO28/ADC2	34	GP28
GND	8	GND	AGND	33	GND
GP6	9	GPIO6	IO27/ADC1	32	GP27
GP7	10	GPIO7	IO26/ADC0	31	GP26
GP8	11	GPIO8	RUN	30	RUN
GP9	12	GPIO9	GPIO22	29	GP22
GND	13	GND	GND	28	GND
GP10	14	GPIO10	GPIO21	27	GP21
GP11	15	GPIO11	GPIO20	26	GP20
GP12	16	GPIO12	GPIO19	25	GP19
GP13	17	GPIO13	GPIO18	24	GP18
GND	18	GND	GND	23	GND
GP14	19	GPIO14	GPIO17	22	GP17
GP15	20	GPIO15	GPIO16	21	GP16

DS3231 RTC Connections

U2: DS3231

Pin	Function	Connection
1	32KHz	NC
2	VCC	+3.3V
3	INT/SQW	INT (via R2 10K)
4	RST	NC
5	NC	NC
6	NC	NC
7	NC	NC
8	NC	NC
9	NC	NC
10	NC	NC
11	NC	NC
12	NC	NC
13	VBAT	BAT1
14	GND	GND
15	P SDA	GP20
16	P SCL	GP21

Power and Signal Connections

Power Connections:

- 3V3(OUT):** Connected to +3.3V.
- GP3:** Connected to 3V3_EN via R5 (0R/NC).
- 3V3 EN:** Connected to 3V3(OUT) via R6 (0R/NC).
- RUN:** Connected to 3V3(OUT) via R7 (0R/NC).

Signal Connections:

- INT:** Connected to GP28 via R2 (10K).
- P SCL:** Connected to GP21 via R3 (10K/NC).
- P SDA:** Connected to GP20 via R4 (10K/NC).

LED Connection

L_PWR1: Connected to GP17 via R1 (1K) and R8 (0R/NC).

Waveshare Logo

Waveshare

Pico-RTC-DS3231

