TUT-ESP-001: Arduino on EmStat Pico dev board

Last document update: 27-3-2019

Tutorial

This document describes how to connect an Arduino MKR Zero to the EmStat Pico development board.

Contents

1 Getting Started

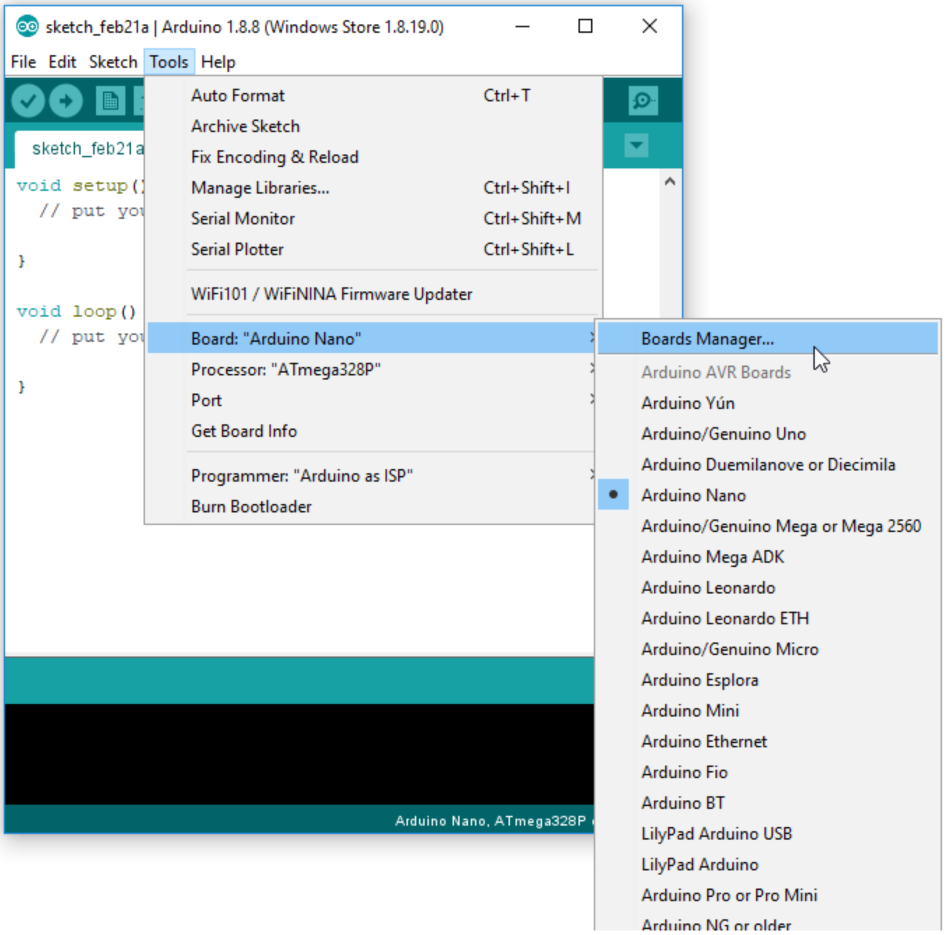
1.1 Hardware Requirements

- Emstat Pico Development Kit
- Arduino MKR Family
- Micro USB cable

1.2 Software Installation

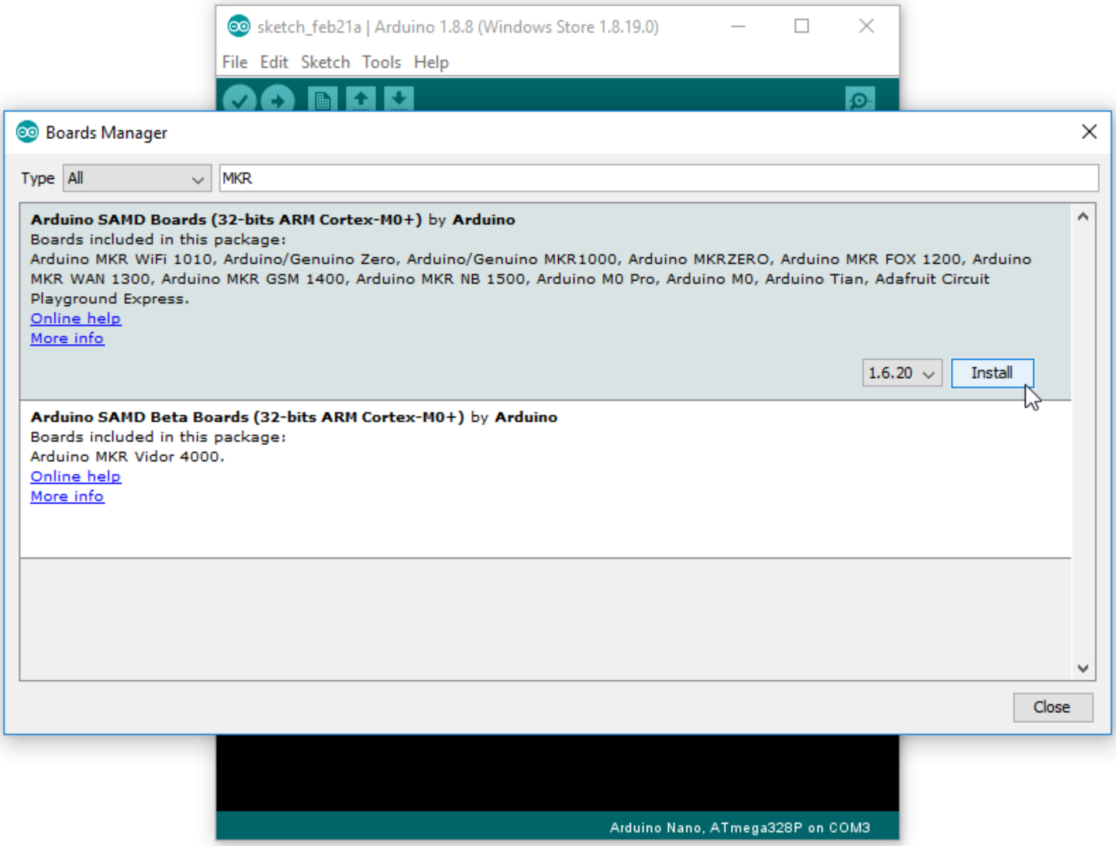
Before starting with the Pico Development board in combination with the Arduino MKR the Arduino software has to be downloaded. The software is freely obtainable on the official website of Arduino https://www.arduino.cc. We advise to always download the most recent version and always download it from the official site.
Once the software has been installed. Open the application and follow the step shown.

Step 1. Select " Tools -> Board: -> Board Manager " as shown.



```
void setup()
  // put you
}

void loop()
  // put you
}
```

Step 2. After Step one, a pop-up should be available. In the empty search bar, type MKR. If everything went accordingly there should be two options. Now install the one where the packages include the MKR series as illustrated underneath.


sketch_feb21a | Arduino 1.8.8 (Windows Store 1.8.19.0)
File Edit Sketch Tools Help
Boards Manager
Online help
More info
1.6.20
Install
Online help
More info
Close

Type All MKR

Arduino SAMD Boards (32-bits ARM Cortex-M0+) by **Arduino**
Boards included in this package:
Arduino MKR WiFi 1010, Arduino/Genuino Zero, Arduino/Genuino MKR1000, Arduino MKRZERO, Arduino MKR FOX 1200, Arduino MKR WAN 1300, Arduino MKR GSM 1400, Arduino MKR NB 1500, Arduino M0 Pro, Arduino M0, Arduino Tian, Adafruit Circuit Playground Express.

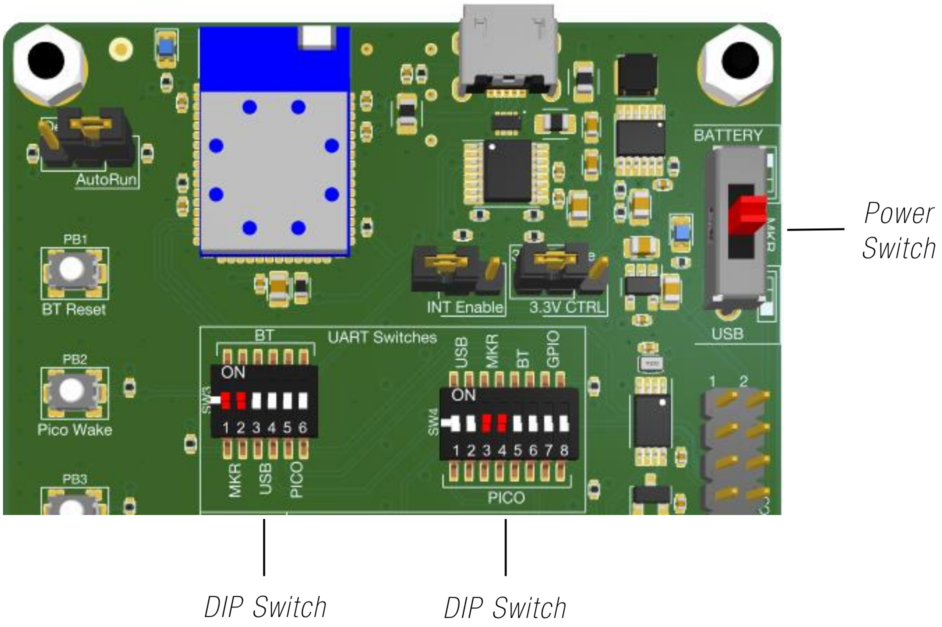
Arduino SAMD Beta Boards (32-bits ARM Cortex-M0+) by **Arduino**
Boards included in this package:
Arduino MKR Vidor 4000.

Arduino Nano, ATmega328P on COM3

Step 3. When installed, close the pop-up menu. The necessary software to control the Arduino and compile new code is installed.

2 Example: UART Communications

There are two usable UART connections between the Arduino MKR and the Pico Development board. One UART connection is connected to the Bluetooth and the other is connected to the Emstat Pico. They can both be used at the same time. For this example the Arduino MKR is going to be the heart of the UART communication. Via Bluetooth we send the character ***t*** with LF (line feed, \n). The Bluetooth module will receive the character and transmit this over UART to the Arduino MKR. The Arduino MKR will forward this character over UART to the Emstat Pico. Then the Emstat Pico will answer with its version, through the MKR to the Bluetooth and back to the device that has send the character ***t***.



2.1 Board Settings

To interact with the Bluetooth module and Emstat Pico via the Arduino MKR. The following setting on the Pico Development board have to be made. All board settings needed for these steps are illustrated in above image marked “RED”.

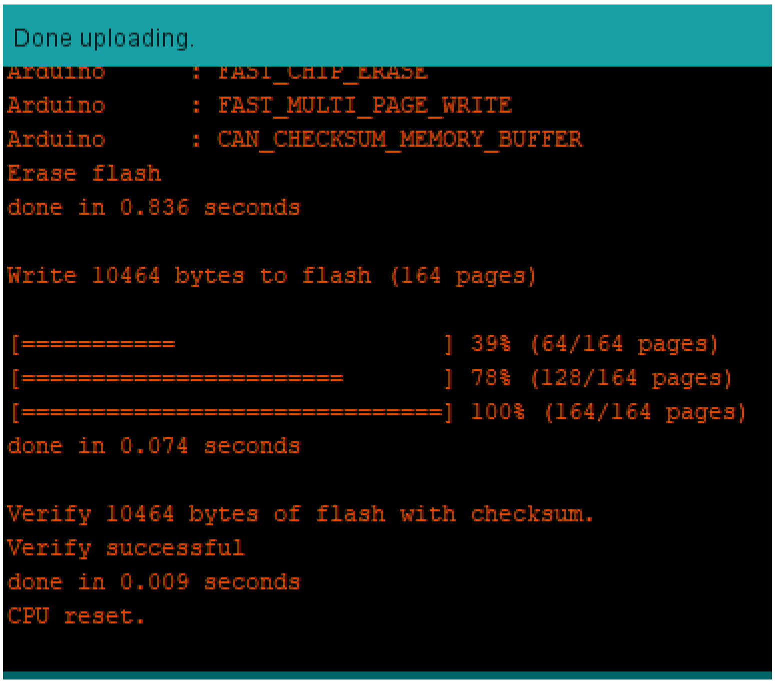
1. The left two switches (1, 2) on SW3 need to be set in the ON position (Top).
2. The left (middle) two switches (3, 4) on SW4 need to be set in the ON position (Top).
3. Connect the USB cable to the USB connector of the Arduino MKR
4. Power-up the board by settings the power switch to MKR (Middle)

2.2 Firmware

This example will assume that the original Emstat Pico Bluetooth script is running on the Bluetooth and the original Pico firmware is running on the Pico. If not, the Bluetooth should be programmed with the correct firmware by following the *Getting started with Bluetooth* Tutorial. For the Pico the Firmware should be flashed via the newest version of PSTrace.

2.3 Programming the Arduino

Open the Arduino application and "File -> Open -> *MkrUart_Example.ino*". If the ino file is loaded and the MKR is connected through USB to PC then Sketch -> Upload. If everything went OK there should be an output similar to the image underneath.



Done Uploading

CPU Reset

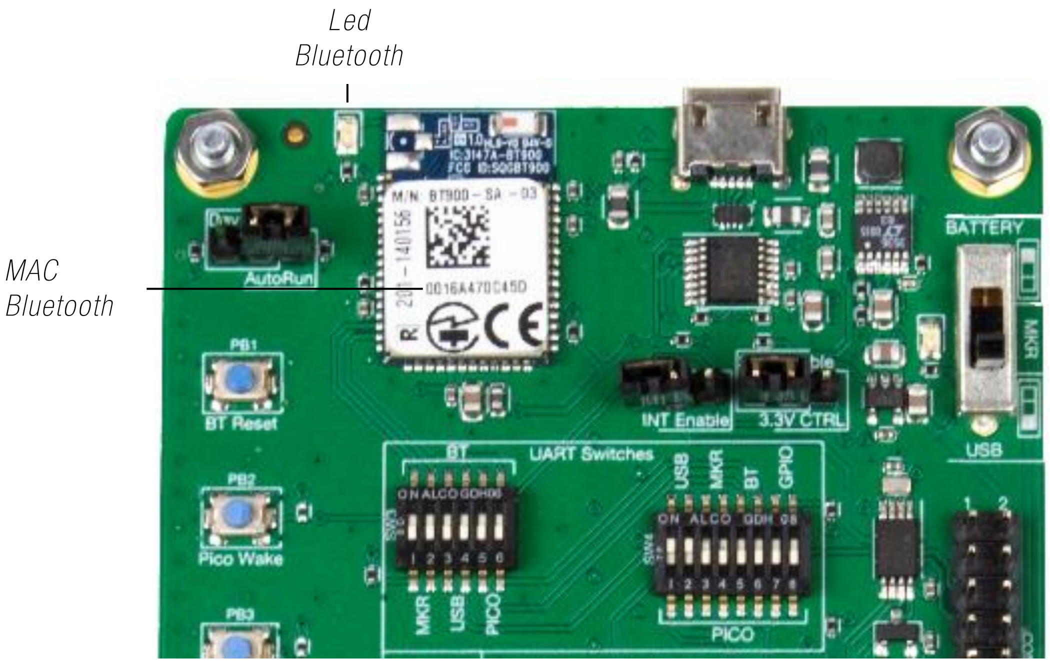
Done uploading.

```
Arduino       : FAST_CHIP_ERASE
Arduino       : FAST_MULTI_PAGE_WRITE
Arduino       : CAN_CHECKSUM_MEMORY_BUFFER
Erase flash
done in 0.836 seconds

Write 10464 bytes to flash (164 pages)

[============                  ] 39% (64/164 pages)
[=======================       ] 78% (128/164 pages)
[==============================] 100% (164/164 pages)
done in 0.074 seconds

Verify 10464 bytes of flash with checksum.
Verify successful
done in 0.009 seconds
CPU reset.
```

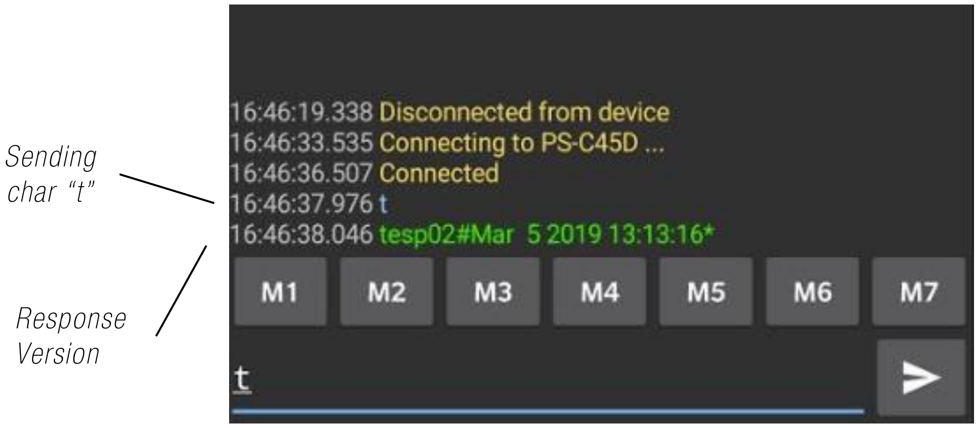
If something went wrong follow the Error section.

2.4 Results

To see is everything went accordingly a terminal program can be used on a device with Bluetooth. We tested this with the Android application *Serial Bluetooth Terminal* from *Kai Morich.* First we have paired the Bluetooth device with our Andriod device. The Emstat Pico Development board wil indentify itself with PS-_ _ _ _ (four characters). These four characters are the last four charaters of the Bluetooth MAC adress. You can find this on the Bluetooth module on the development board as shown underneath. In our exemple it is PS-C45D.



Once connected with the Emstat Pico Development board, as shown underneath, the Bluetooth LED indicator should go ON. Now a character *t* can be send to get the version of the Emstat Pico as shown underneath.

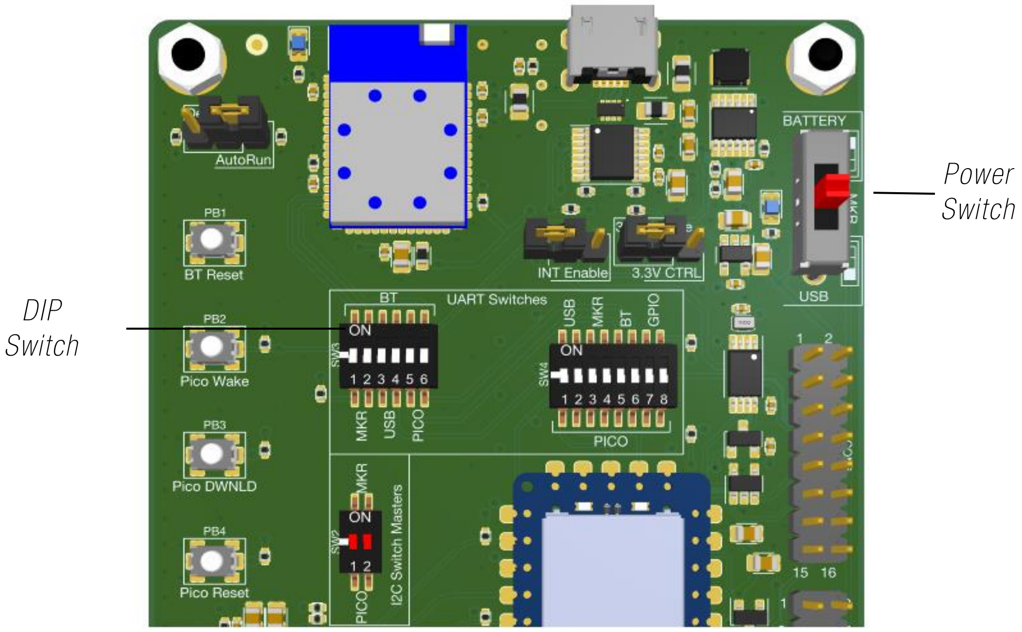

```
16:46:19.338 Disconnected from device
16:46:33.535 Connecting to PS-C45D ...
16:46:36.507 Connected
16:46:37.976 t
16:46:38.046 tesp02#Mar  5 2019 13:13:16*
```

3 Example: I2C Communications

This example will write "*Hello Emstat Pico Dev*" to the Eeprom and read the characters back. The main example revolves around two functions. One function is to write bytes to the Eeprom and the other function read bytes from Eeprom.

3.1 Board Settings

There are two I2C slave devices on the Development board available. The Realtime Clock (RTC) and the Eeprom. Both slave devices can be controlled by either the Arduino MKR or the Emstat Pico depending on the position of switch SW2. This switch will enable one of the two devices as Master. In this example the Arduino MKR is selected as master.

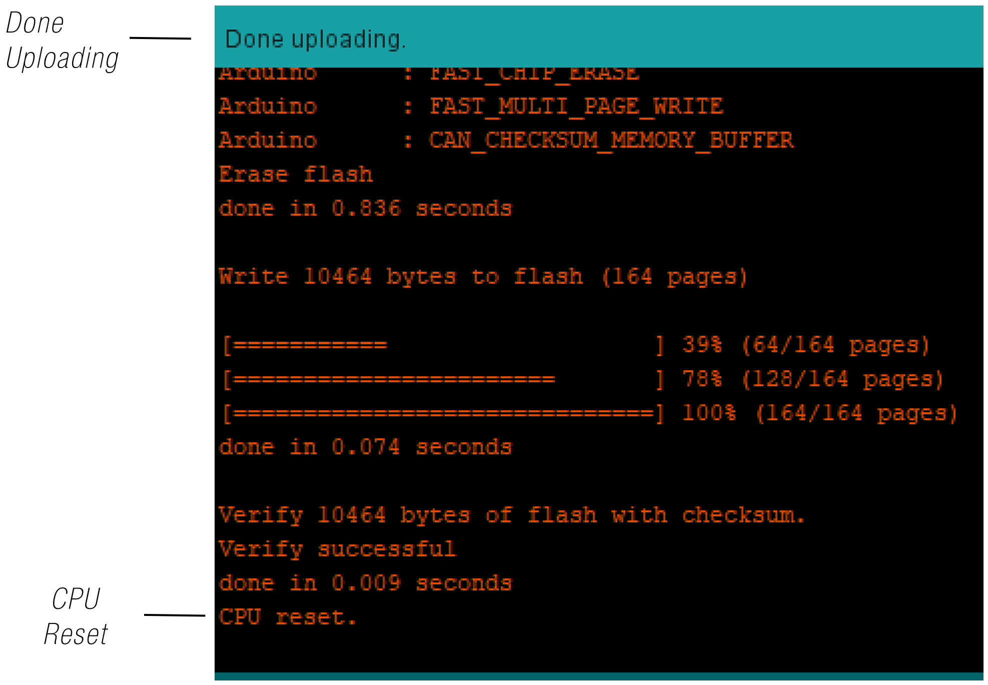


To interact with the RTC and Eeprom via the Arduino MKR. The following setting on the Pico Development board have to be made. All board settings needed for these steps are illustrated in above image marked "RED".

1. The two switches on SW2 need to be set in the ON position (Top).
2. Connect the USB cable to the USB connector of the Arduino MKR
3. Power-up the board by settings the power switch to MKR (Middle)

3.2 Programming The Arduino

Open the Arduino application and "File -> Open -> *Eeprom_Example.ino" .* If the ino file is loaded and the MKR is connected through USB to PC then Sketch -> Upload. If everything went OK there should be an output similar to the image underneath.

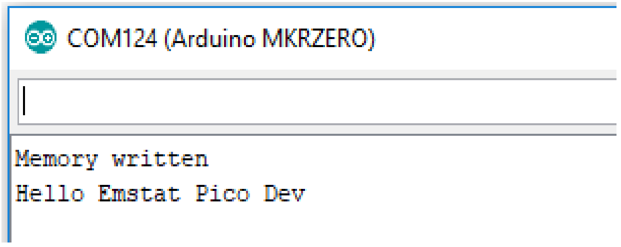


```
Done uploading.
Arduino       : FAST_CHIP_ERASE
Arduino       : FAST_MULTI_PAGE_WRITE
Arduino       : CAN_CHECKSUM_MEMORY_BUFFER
Erase flash
done in 0.836 seconds

Write 10464 bytes to flash (164 pages)

[============                  ] 39% (64/164 pages)
[========================      ] 78% (128/164 pages)
[==============================] 100% (164/164 pages)
done in 0.074 seconds

Verify 10464 bytes of flash with checksum.
Verify successful
done in 0.009 seconds
CPU reset.
```

If Something went wrong follow the Error section.

3.3 Results

To see is everything went accordingly a serial monitor can be opened from the Arduino Application "Tools -> Serial Monitor *or the shortcut Ctrl+Shift+m.* The following output can be seen in the Serial Monitor. The "*Hello Emstat Pico Dev*" will loop infinitely.



```
Memory written
Hello Emstat Pico Dev
```

4 Example: Realtime Clock (RTC)

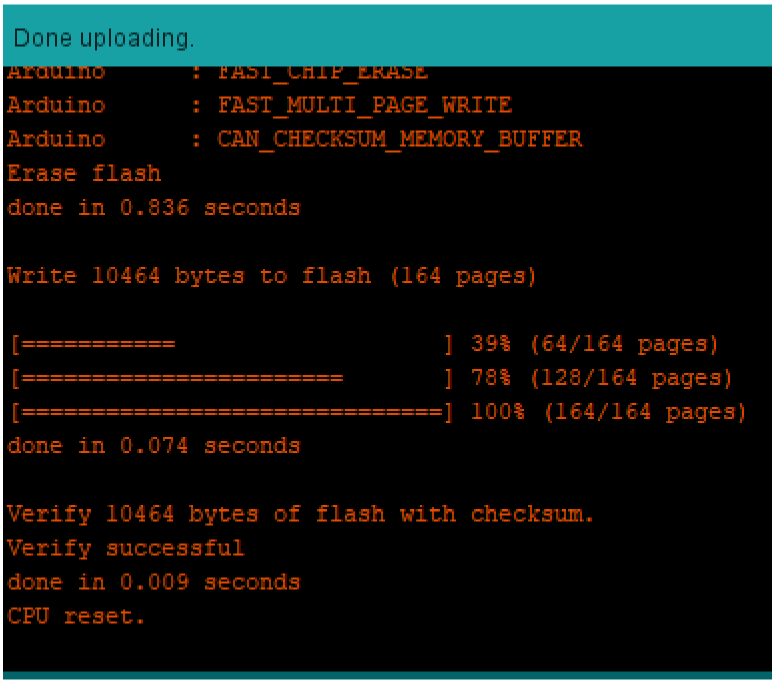
This example will write the Time of Day to the Serial Monitor. In this example the alarm and time can be set. Once the Time of day is equal to the alarm time an interrupt 1 will go off. If the Jumper INT Enable is connected. The Interrupt from the RTC will be sent to the Pico's "GPIO_7/*Wake*" pin (24). This will enable the Pico to come out of hibernation.

4.1 Board Settings

See section 0.

4.2 Programming the Arduino

Open the Arduino application and "File -> Open -> *PicoRTC_Example.ino" .* If the ino file is loaded and the MKR is connected through USB to PC then Sketch -> Upload. If everything went OK there should be an output similar to the image underneath.



Done Uploading

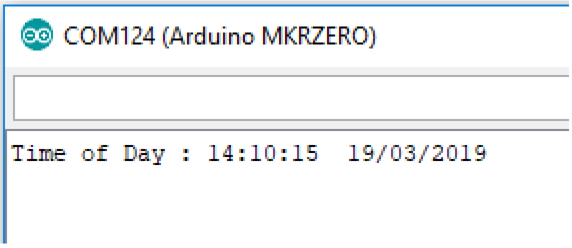
CPU Reset

```
Done uploading.
Arduino       : FAST_CHIP_ERASE
Arduino       : FAST_MULTI_PAGE_WRITE
Arduino       : CAN_CHECKSUM_MEMORY_BUFFER
Erase flash
done in 0.836 seconds

Write 10464 bytes to flash (164 pages)

[===========                   ] 39% (64/164 pages)
[=======================       ] 78% (128/164 pages)
[==============================] 100% (164/164 pages)
done in 0.074 seconds

Verify 10464 bytes of flash with checksum.
Verify successful
done in 0.009 seconds
CPU reset.
```

If something went wrong follow the Error section.

4.3 Results

To see is everything went accordingly a serial monitor can be opened from the Arduino Application "Tools -> Serial Monitor *or the shortcut Ctrl+Shift+m.* The following output can be seen in the Serial Monitor. The "*Time of Day : ….* " will loop infinity.



```
Time of Day : 14:10:15  19/03/2019
```

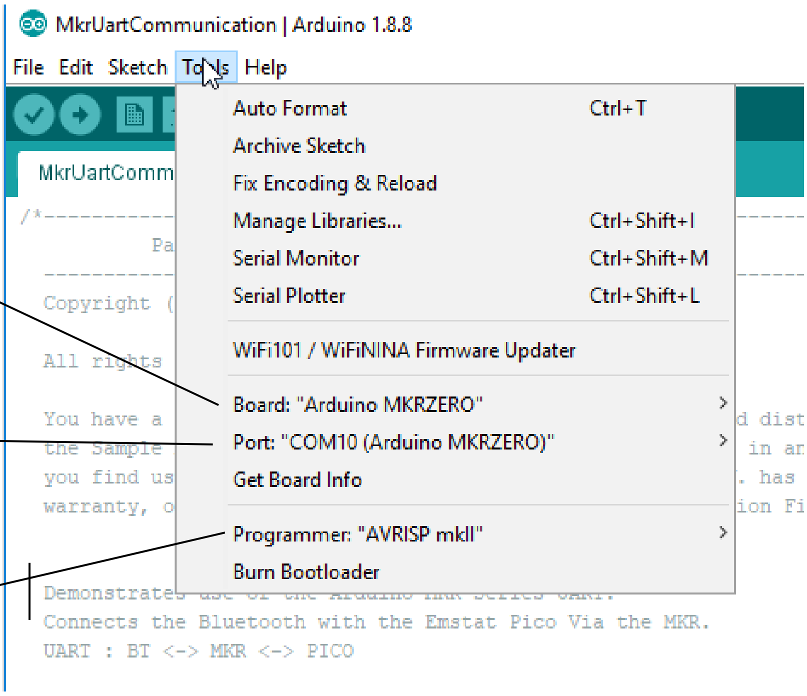
5 Arduino Error

Errors often have to do with the COM port selection. See in the Arduino application if under *Tools* all the settings are correct.

Check whether the correct Arduino is selected

Is the right COM port selected

Check the Programmer


MkrUartCommunication | Arduino 1.8.8
File
Edit
Sketch
Tools
Help
MkrUartComm
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino MKRZERO"
Port: "COM10 (Arduino MKRZERO)"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader

```
Connects the Bluetooth with the Emstat Pico Via the MKR.
UART : BT <-> MKR <-> PICO
```

When this didn't solve anything, try to reset the Arduino either by pushing a button or by disconnection the Arduino from the USB so that the board will power down.