# Getting Familiar with Arduino Nano 33, Raspberry Pi Pico, ArduCAM Camera, TensorFlow Lite

## Arduino Nano 33 and TensorFlow Lite

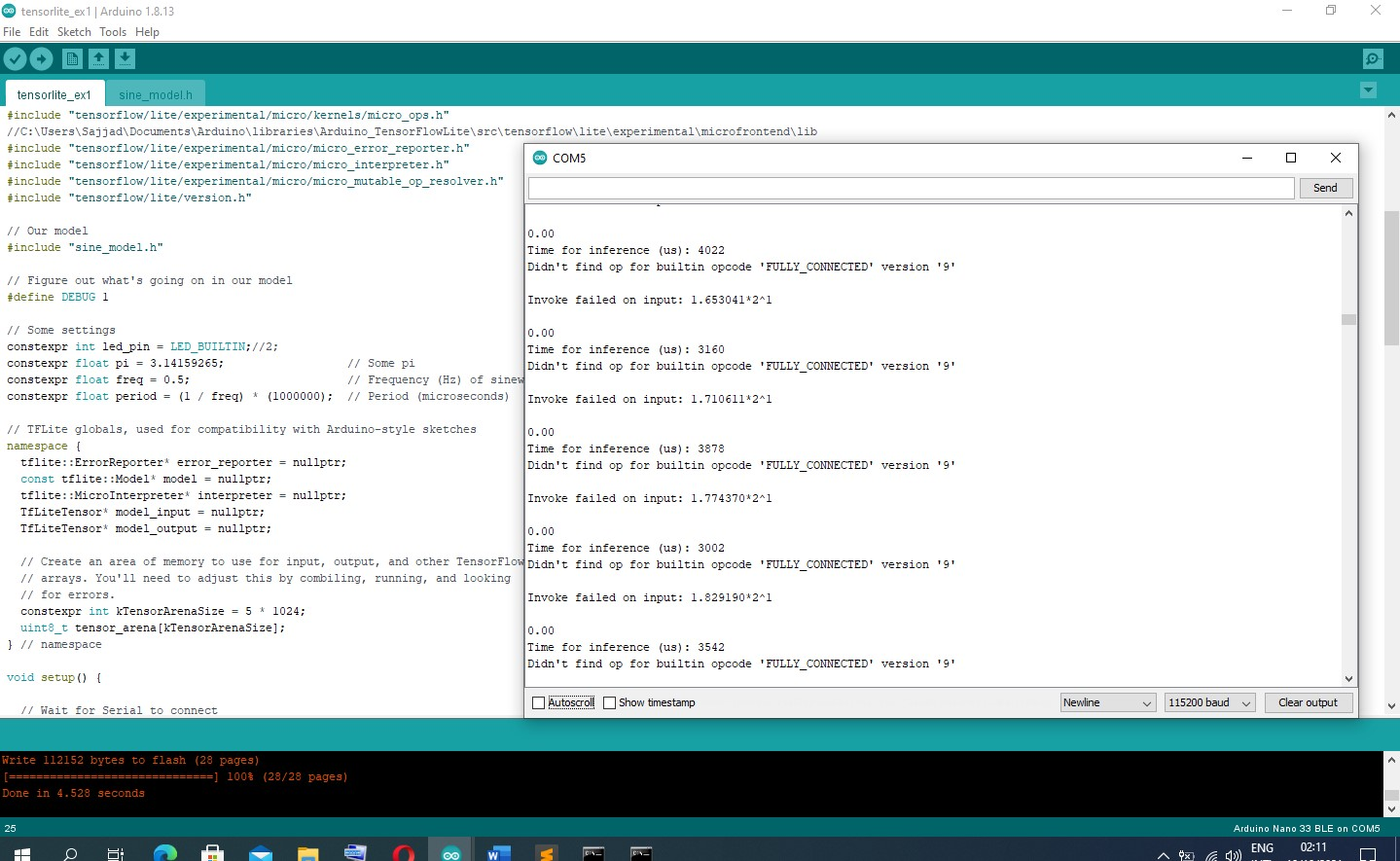
### Basics of TensorFlow

NOC:Practical Machine Learning with TensorFlow

[https://nptel.ac.in/courses/106/106/106106213/#](https://nptel.ac.in/courses/106/106/106106213/)

### Predicting a Sine Value

1. From the example, google-colab **tflite\_sinewave\_training.ipynb**, I generated the sine\_model.h and wrote a Arduino sketch tensorlite\_ex1.ino.
2. I configured and tried to run the example Tensor Lite program, but got this message “Didn't find op for builtin opcode 'FULLY\_CONNECTED' version '9'”. I installed the Tensor Lite version 1.0.15-Alpha, which was mentioned in the example.



1. I started debugging and reading about the error.
2. I used the Arduino example, which is the same as we are doing it. “hello\_world.ino” is similar to tensorlite\_ex1.ino but it has its pre-generated model header file. It was executed without any errors.
3. The values seen on serial port and serial plotter are seems to be a sinusoidal waveform. Which is correct.
4. Then, I fixed that the data being generated from our python is somehow not correct.
5. Then found that somehow, some parameters are to be changed as below:

##converter.optimizations = [tf.lite.Optimize.OPTIMIZE\_FOR\_SIZE]

#converter.optimizations = [tf.lite.Optimize.DEFAULT]

#tflite\_model = converter.convert()

#open(tflite\_model\_name + '.tflite', 'wb').write(tflite\_model)

#This is giving following error while sine\_model.h is used in Arduino

#Didn't find op for builtin opcode 'FULLY\_CONNECTED' version '9'

# following code resolved the issue

#https://stackoverflow.com/questions/63620419/firebase-local-model-throws-didnt-find-op-for-builtin-opcode-conv-2d-version

converter = tf.lite.TFLiteConverter.from\_keras\_model(model)

## ArduCAM Camera

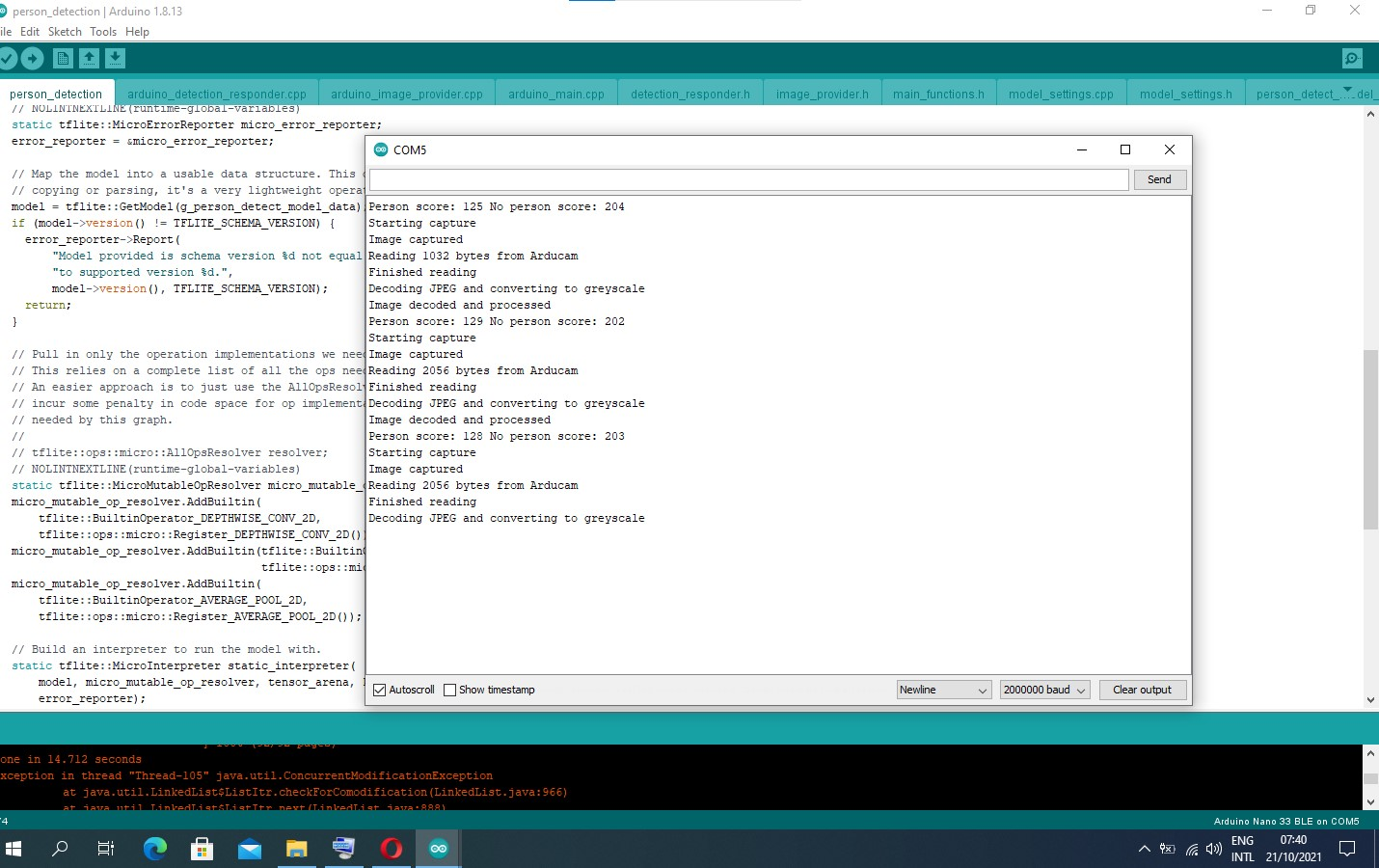
1. ArduCAM-M-2MP Camera Shield was connected with Arduino Nano 33. The Camera was not detected, I debugged, but could not resolve.

|  |  |
| --- | --- |
| **ArduCAM** | **Arduino** |
| CS | D10 |
| MOSI | D11 |
| MISO | D12 |
| SCK | D13 |
| GND | GND |
| VCC | +5V |
| SDA | A4 |
| SCL | A5 |

1. Also programmed the “ArduCAM\_Mini\_2MP\_OV2640\_functions.ino” and tried to interact with the camera using “ArduCAM\_Host\_V2.0\_Windows.exe” application. It was giving SPI error.
2. Then I connected VCC of the Camera to the 3V3 of the Arduino and it works now.
3. I used “ArduCAM\_Mini\_2MP\_OV2640\_functions.ino” and then took some pictures and videos using “ArduCAM\_Host\_V2.0\_Windows.exe”.

### Person Detection Example

1. Programmed “person\_detection.ino” and then get the running algorithm.



## Raspberry Pi Pico

1. I have spent hours installing and checking the Pi Pico Toolchain to run an example program.
2. It was successfully installed and can run from command prompt.
3. I can't detect it when plugged in to different laptops and also, I checked with TECO HW lab. There is a beep when I plug in the Pico, but there is no derive popped on the computer.
4. Later, I found that, while inserting the USB to the PC, the Pi Pico RESET button should be pressed.

### Setup pico-sdk and pico-examples using the command prompt and successfully run the blink example on RPI-Pico.

D:\rpi-pico\pico-examples> mkdir build

D:\rpi-pico\pico-examples> cd build

D:\rpi-pico\pico-examples\build>cmake -G "NMake Makefiles" ..

PICO\_SDK\_PATH is D:/rpi-pico/pico-sdk

PICO platform is rp2040.

PICO target board is pico.

Using board configuration from D:/rpi-pico/pico-sdk/src/boards/include/boards/pico.h

TinyUSB available at D:/rpi-pico/pico-sdk/lib/tinyusb/src/portable/raspberrypi/rp2040; adding USB support.

-- Configuring done

-- Generating done

-- Build files have been written to: D:/rpi-pico/pico-examples/build

D:\rpi-pico\pico-examples\build>nmake

…

…

D:\rpi-pico\pico-examples\build>

### Building your own project and implementing using the command prompt and successfully run the project example on RPI-Pico.

D:\rpi-pico>mkdir pico-empty-project

D:\rpi-pico>cd pico-empty-project

D:\rpi-pico\pico-empty-project>mkdir build

D:\rpi-pico\pico-empty-project>mkdir main

D:\rpi-pico\pico-empty-project>create CMakeLists.txt

D:\rpi-pico\pico-empty-project>copy from pico-exampel example\_auto\_set\_url.cmake and pico\_sdk\_import.cmake

D:\rpi-pico\pico-empty-project>vim

'vim' is not recognized as an internal or external command,

operable program or batch file.

D:\rpi-pico\pico-empty-project>vi

'vi' is not recognized as an internal or external command,

operable program or batch file.

D:\rpi-pico\pico-empty-project>dir

Volume in drive D is Hussain

Volume Serial Number is C43F-8639

Directory of D:\rpi-pico\pico-empty-project

21/10/2021 04:05 <DIR> .

21/10/2021 04:05 <DIR> ..

21/10/2021 04:02 <DIR> build

21/10/2021 04:04 351 CMakeLists.txt

18/10/2021 03:54 301 example\_auto\_set\_url.cmake

21/10/2021 04:06 <DIR> main

18/10/2021 03:54 2,825 pico\_sdk\_import.cmake

3 File(s) 3,477 bytes

4 Dir(s) 18,543,718,400 bytes free

D:\rpi-pico\pico-empty-project>cd build

D:\rpi-pico\pico-empty-project\build>cmake -G "NMake Makefiles" ..

Using PICO\_SDK\_PATH from environment ('d:\rpi-pico\pico-sdk')

PICO\_SDK\_PATH is D:/rpi-pico/pico-sdk

Defaulting PICO\_PLATFORM to rp2040 since not specified.

Defaulting PICO platform compiler to pico\_arm\_gcc since not specified.

-- Defaulting build type to 'Release' since not specified.

PICO compiler is pico\_arm\_gcc

-- The C compiler identification is GNU 10.3.1

-- The CXX compiler identification is GNU 10.3.1

-- The ASM compiler identification is GNU

-- Found assembler: C:/Program Files (x86)/GNU Arm Embedded Toolchain/10 2021.07/bin/arm-none-eabi-gcc.exe

Defaulting PICO target board to pico since not specified.

Using board configuration from D:/rpi-pico/pico-sdk/src/boards/include/boards/pico.h

-- Found Python3: C:/Program Files (x86)/Python39-32/python.exe (found version "3.9.5") found components: Interpreter

TinyUSB available at D:/rpi-pico/pico-sdk/lib/tinyusb/src/portable/raspberrypi/rp2040; adding USB support.

-- Configuring done

-- Generating done

-- Build files have been written to: D:/rpi-pico/pico-empty-project/build

D:\rpi-pico\pico-empty-project\build>nmake

Microsoft (R) Program Maintenance Utility Version 14.29.30136.0

Copyright (C) Microsoft Corporation. All rights reserved.

Scanning dependencies of target bs2\_default

[ 1%] Building ASM object pico-sdk/src/rp2\_common/boot\_stage2/CMakeFiles/bs2\_default.dir/compile\_time\_choice.S.obj

[ 3%] Linking ASM executable bs2\_default.elf

[ 3%] Built target bs2\_default

[ 5%] Generating bs2\_default.bin

[ 6%] Generating bs2\_default\_padded\_checksummed.S

[ 6%] Built target bs2\_default\_padded\_checksummed\_asm

[ 8%] Creating directories for 'ELF2UF2Build'

[ 10%] No download step for 'ELF2UF2Build'

[ 11%] No update step for 'ELF2UF2Build'

[ 13%] No patch step for 'ELF2UF2Build'

[ 15%] Performing configure step for 'ELF2UF2Build'

-- The C compiler identification is MSVC 19.29.30136.0

-- The CXX compiler identification is MSVC 19.29.30136.0

-- Detecting C compiler ABI info

-- Detecting C compiler ABI info - done

-- Check for working C compiler: C:/Program Files (x86)/Microsoft Visual Studio/2019/BuildTools/VC/Tools/MSVC/14.29.30133/bin/Hostx86/x86/cl.exe - skipped

-- Detecting C compile features

-- Detecting C compile features - done

-- Detecting CXX compiler ABI info

-- Detecting CXX compiler ABI info - done

-- Check for working CXX compiler: C:/Program Files (x86)/Microsoft Visual Studio/2019/BuildTools/VC/Tools/MSVC/14.29.30133/bin/Hostx86/x86/cl.exe - skipped

-- Detecting CXX compile features

-- Detecting CXX compile features - done

-- Configuring done

-- Generating done

-- Build files have been written to: D:/rpi-pico/pico-empty-project/build/elf2uf2

[ 16%] Performing build step for 'ELF2UF2Build'

Microsoft (R) Program Maintenance Utility Version 14.29.30136.0

Copyright (C) Microsoft Corporation. All rights reserved.

[ 50%] Building CXX object CMakeFiles/elf2uf2.dir/main.cpp.obj

main.cpp

D:\rpi-pico\pico-sdk\tools\elf2uf2\main.cpp(328): warning C4996: 'fopen': This function or variable may be unsafe. Consider using fopen\_s instead. To disable deprecation, use \_CRT\_SECURE\_NO\_WARNINGS. See online help for details.

D:\rpi-pico\pico-sdk\tools\elf2uf2\main.cpp(334): warning C4996: 'fopen': This function or variable may be unsafe. Consider using fopen\_s instead. To disable deprecation, use \_CRT\_SECURE\_NO\_WARNINGS. See online help for details.

[100%] Linking CXX executable elf2uf2.exe

[100%] Built target elf2uf2

[ 18%] No install step for 'ELF2UF2Build'

[ 20%] Completed 'ELF2UF2Build'

[ 20%] Built target ELF2UF2Build

Scanning dependencies of target main

[ 22%] Building C object main/CMakeFiles/main.dir/main.c.obj

[ 23%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/pico\_stdlib/stdlib.c.obj

[ 25%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/hardware\_gpio/gpio.c.obj

[ 27%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/hardware\_claim/claim.c.obj

[ 28%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/pico\_platform/platform.c.obj

[ 30%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/hardware\_sync/sync.c.obj

[ 32%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/hardware\_uart/uart.c.obj

[ 33%] Building ASM object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/hardware\_divider/divider.S.obj

[ 35%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/common/pico\_time/time.c.obj

[ 37%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/common/pico\_time/timeout\_helper.c.obj

[ 38%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/hardware\_timer/timer.c.obj

[ 40%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/common/pico\_sync/sem.c.obj

[ 42%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/common/pico\_sync/lock\_core.c.obj

[ 44%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/common/pico\_sync/mutex.c.obj

[ 45%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/common/pico\_sync/critical\_section.c.obj

[ 47%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/common/pico\_util/datetime.c.obj

[ 49%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/common/pico\_util/pheap.c.obj

[ 50%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/common/pico\_util/queue.c.obj

[ 52%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/pico\_runtime/runtime.c.obj

[ 54%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/hardware\_clocks/clocks.c.obj

[ 55%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/hardware\_irq/irq.c.obj

[ 57%] Building ASM object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/hardware\_irq/irq\_handler\_chain.S.obj

[ 59%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/hardware\_pll/pll.c.obj

[ 61%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/hardware\_vreg/vreg.c.obj

[ 62%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/hardware\_watchdog/watchdog.c.obj

[ 64%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/hardware\_xosc/xosc.c.obj

[ 66%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/pico\_printf/printf.c.obj

[ 67%] Building ASM object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/pico\_bit\_ops/bit\_ops\_aeabi.S.obj

[ 69%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/pico\_bootrom/bootrom.c.obj

[ 71%] Building ASM object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/pico\_divider/divider.S.obj

[ 72%] Building ASM object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/pico\_double/double\_aeabi.S.obj

[ 74%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/pico\_double/double\_init\_rom.c.obj

[ 76%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/pico\_double/double\_math.c.obj

[ 77%] Building ASM object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/pico\_double/double\_v1\_rom\_shim.S.obj

[ 79%] Building ASM object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/pico\_int64\_ops/pico\_int64\_ops\_aeabi.S.obj

[ 81%] Building ASM object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/pico\_float/float\_aeabi.S.obj

[ 83%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/pico\_float/float\_init\_rom.c.obj

[ 84%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/pico\_float/float\_math.c.obj

[ 86%] Building ASM object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/pico\_float/float\_v1\_rom\_shim.S.obj

[ 88%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/pico\_malloc/pico\_malloc.c.obj

[ 89%] Building ASM object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/pico\_mem\_ops/mem\_ops\_aeabi.S.obj

[ 91%] Building ASM object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/pico\_standard\_link/crt0.S.obj

[ 93%] Building CXX object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/pico\_standard\_link/new\_delete.cpp.obj

[ 94%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/pico\_standard\_link/binary\_info.c.obj

[ 96%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/pico\_stdio/stdio.c.obj

[ 98%] Building C object main/CMakeFiles/main.dir/D\_/rpi-pico/pico-sdk/src/rp2\_common/pico\_stdio\_uart/stdio\_uart.c.obj

[100%] Linking CXX executable main.elf

[100%] Built target main

D:\rpi-pico\pico-empty-project\build>

### Setup pico-sdk and pico-examples using the Visual Studio Code and successfully run the blink example on RPI-Pico.

1. Copied fresh copy of pico-example into pico-exmaple1
2. created a build directory inside pico-examples1
3. started visual studio code from VS2019 Developer command prompt
4. Cmake: Configure Environment

Item: PICO\_SDK\_PATH Value: D:\rpi-pico\pico-sdk

Cmake: Generator

NMake Makefiles

Cmake: Cmake Path

cmake

1. open the folder pico-example1
2. VS Code will automatically configure the project.
3. after that click "build" button at the bottom
4. The VSC bottom looks like:



1. When the folder is first opened

[variant] Loaded new set of variants

[kit] Successfully loaded 6 kits from C:\Users\Sajjad\AppData\Local\CMakeTools\cmake-tools-kits.json

[proc] Executing command: "C:\Program Files (x86)\GNU Arm Embedded Toolchain\10 2021.07\bin\arm-none-eabi-gcc.exe" -v

[main] Configuring folder: pico-examples2

[proc] Executing command: "C:\Program Files\CMake\bin\cmake.EXE" --no-warn-unused-cli -DCMAKE\_EXPORT\_COMPILE\_COMMANDS:BOOL=TRUE -DCMAKE\_BUILD\_TYPE:STRING=Debug "-DCMAKE\_C\_COMPILER:FILEPATH=C:\Program Files (x86)\GNU Arm Embedded Toolchain\10 2021.07\bin\arm-none-eabi-gcc.exe" "-DCMAKE\_CXX\_COMPILER:FILEPATH=C:\Program Files (x86)\GNU Arm Embedded Toolchain\10 2021.07\bin\arm-none-eabi-g++.exe" -Hd:/rpi-pico/pico-examples2 -Bd:/rpi-pico/pico-examples2/build -G "NMake Makefiles"

[cmake] Not searching for unused variables given on the command line.

[cmake] Using PICO\_SDK\_PATH from environment ('D:\rpi-pico\pico-sdk')

[cmake] PICO\_SDK\_PATH is D:/rpi-pico/pico-sdk

[cmake] Defaulting PICO\_PLATFORM to rp2040 since not specified.

[cmake] Defaulting PICO platform compiler to pico\_arm\_gcc since not specified.

[cmake] PICO compiler is pico\_arm\_gcc

[cmake] -- The C compiler identification is GNU 10.3.1

[cmake] -- The CXX compiler identification is GNU 10.3.1

[cmake] -- The ASM compiler identification is GNU

[cmake] -- Found assembler: C:/Program Files (x86)/GNU Arm Embedded Toolchain/10 2021.07/bin/arm-none-eabi-gcc.exe

[cmake] Using regular optimized debug build (set PICO\_DEOPTIMIZED\_DEBUG=1 to de-optimize)

[cmake] Defaulting PICO target board to pico since not specified.

[cmake] Using board configuration from D:/rpi-pico/pico-sdk/src/boards/include/boards/pico.h

[cmake] -- Found Python3: C:/Program Files (x86)/Python39-32/python.exe (found version "3.9.5") found components: Interpreter

[cmake] TinyUSB available at D:/rpi-pico/pico-sdk/lib/tinyusb/src/portable/raspberrypi/rp2040; adding USB support.

[cmake] Compiling TinyUSB with CFG\_TUSB\_DEBUG=1

[cmake] -- Configuring done

[cmake] -- Generating done

[cmake] -- Build files have been written to: D:/rpi-pico/pico-examples2/build

1. When the “build” button at the bottom bar is pressed

[main] Building folder: pico-examples1

…

…

[build] [100%] Built target hello\_watchdog

[build] Build finished with exit code 0

### Building your own project and implementing using the Visual Studio Code and successfully run the project example on RPI-Pico.

1. create a folder called myblinkvsc
2. create your main.c file
3. create CMakeLists file
4. started visual studio code from VS2019 Developer command prompt
5. open the folder myblinkvsc
6. VS Code will automatically configure the project.
7. after that click "build" button at the bottom
8. There was an error while building the projects, then after a through google surfing found that the an older CMake version should be used. There I uninstalled the latest version and installed CMake 3.20.6. (**AR10B2~1.EXE: error: ´╗┐CMakeFiles/hello\_serial.dir/hello\_serial.c.obj: No such file or directory**)