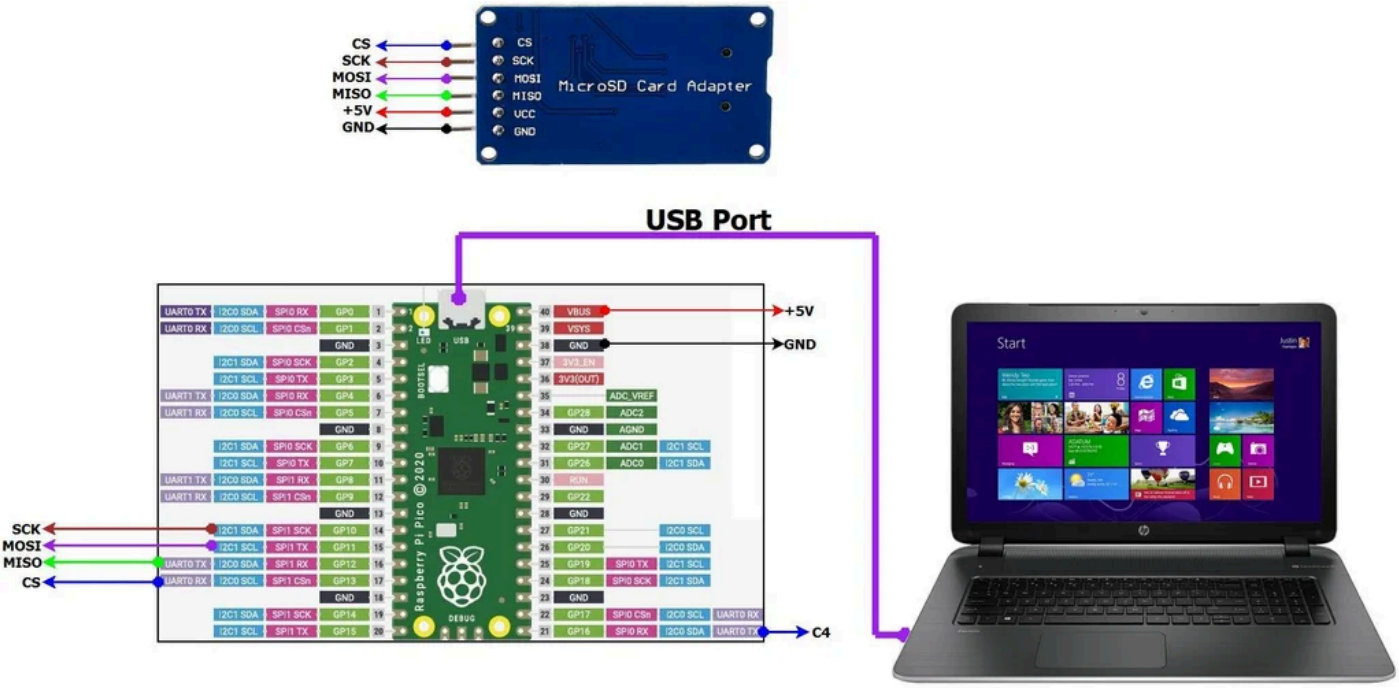
AUTODESK
Instructables

Raspberry Pi Pico -- Micro SD Card Interface

By PugazhM in CircuitsRaspberry Pi

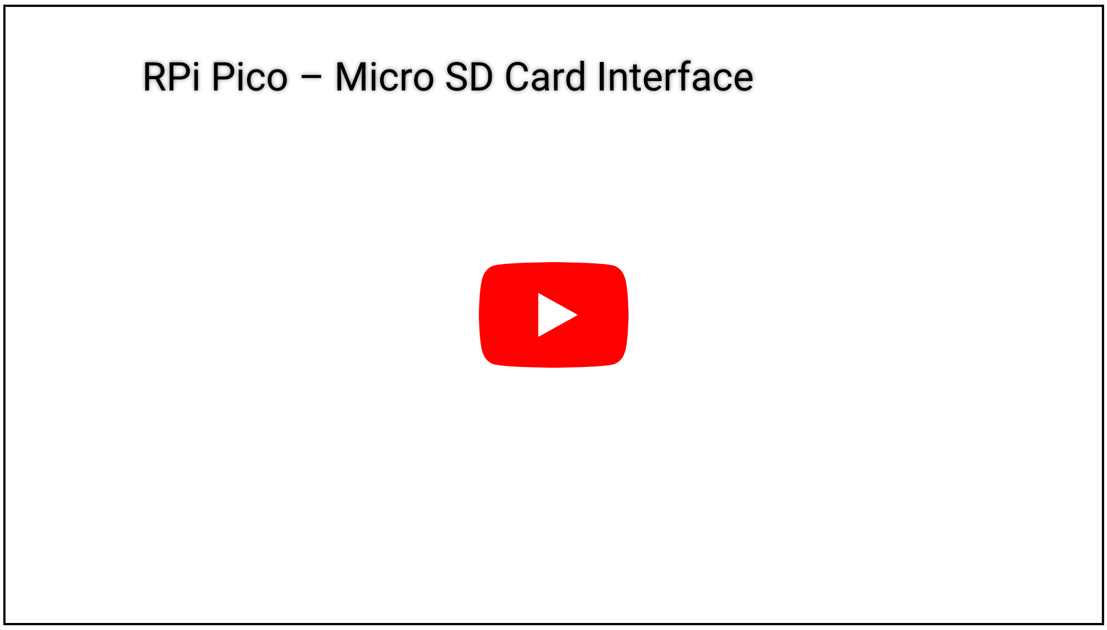
Introduction: Raspberry Pi Pico -- Micro SD Card Interface



In this experiment, a "sample text file" is created inside the SD card, and then written with the running text. RPi Pico opens the “sample text file” for reading and dump the text on debug port.

Visit the "VESZLE - The Art Of Electronics" you tube channel for further information and videos.

https://www.youtube.com/channel/UCY4mekPLfeFinbQHp...



Step 1: Abstract

The Raspberry Pi Pico is a tiny, fast, and versatile board built using RP2040 features a dual-core Arm Cortex-M0+ processor with 264KB internal RAM and support for up to 16MB of off-chip Flash. It provides wide range of flexible I/O options includes I2C, SPI, and uniquely Programmable I/O (GPIO) pins.

Secure Digital Card, abbreviated as SD Card is a non-volatile memory card, commonly used to write and read large quantities of data in smart devices. These days SD cards are available with 4GB to 128GB memory size.

The SD card module provides the micro memory card interface and it is connected to RPi Pico via SPI port. This experiment, creates a "sample text file" inside the SD card, and then written with the running text. RPi Pico opens the "sample text file" for reading and dump the text on debug port.

Step 2: Reference

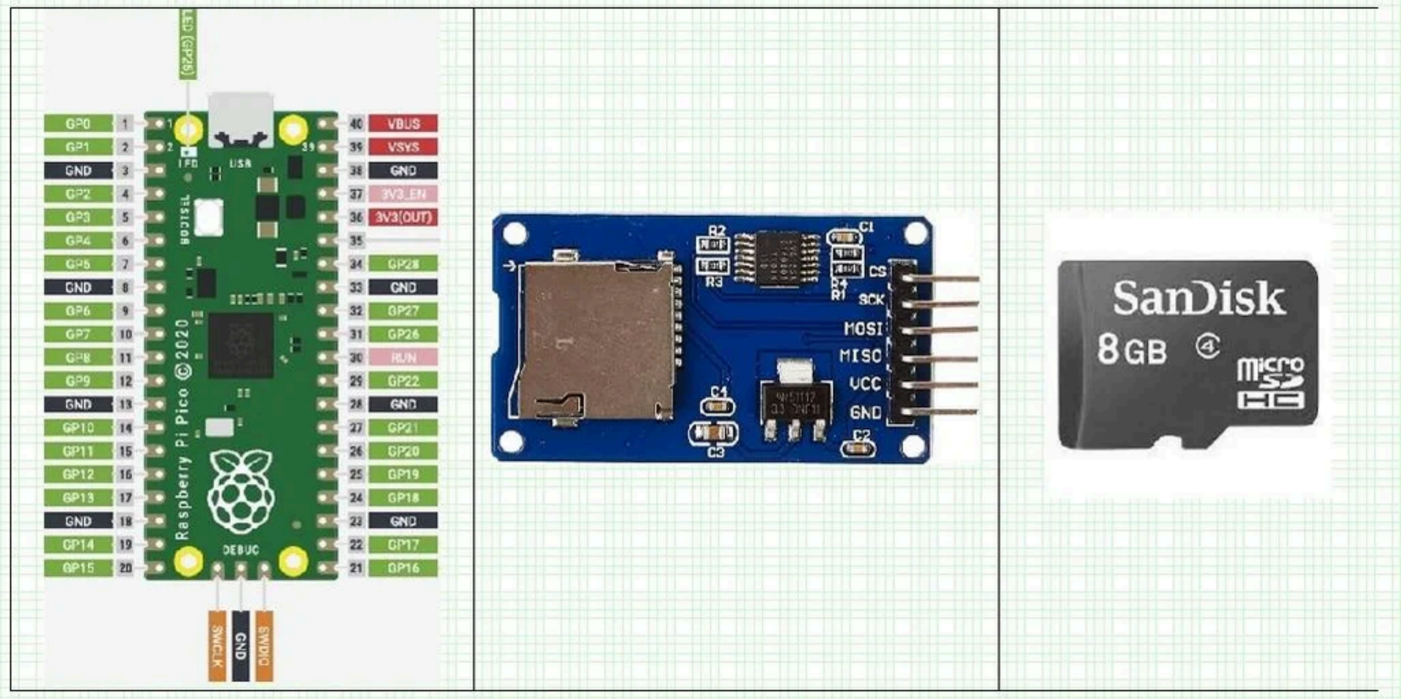
Raspberry Pi Pico, Getting Started on Board Blink LED" Instruct-able / You-tube by PugazhM

https://www.instructables.com/Raspberry-Pi-Pico-G...

The MicroPython "sdcard.py"

https://github.com/micropython/micropython/blob/m...

Step 3: Component



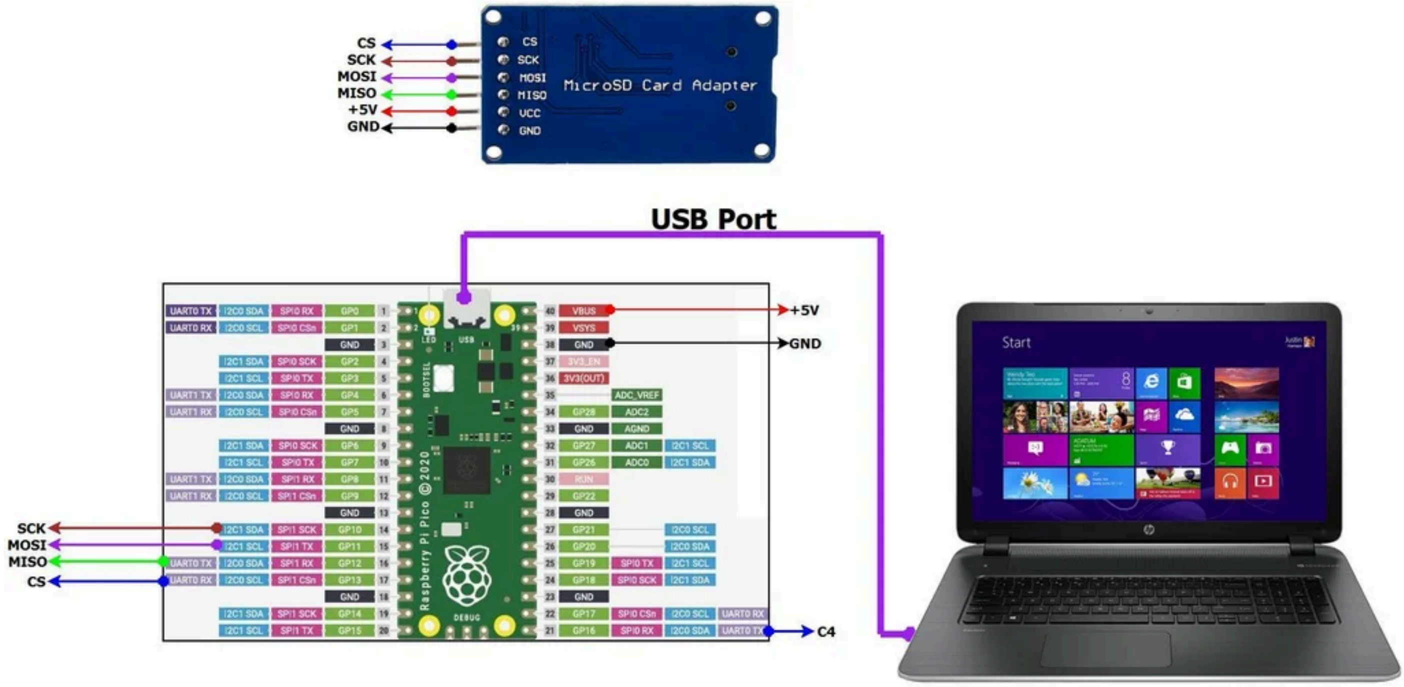
Raspberry Pi Pico

Micro USB Cable

SD Card Module = 1 No

8GB micro-SD card = 1 No

Step 4: Schematic



SD Card module provides “micro-SD card socket”, for inserting the memory card and then provides SPI interface pins (MOSI = Pin11, MISO = Pin12, SCLK = 10 and CS = Pin13), for connecting it into RPi Pico board.

SD Card module converts the +5VDC supply into 3.3VDC using AM1117, 3.3VDC regulator and converts the 5VDC logic Pins into 3.3VDC logic pins by using “74ABT125PW, Quad Buffer Tri state logic converter”.

The micro-SD card should be powered and operated with 3.3VDC. Otherwise, connecting the 5VDC logic into microSD card, can permanently damage the memory card.

This experiment uses an 8GB micro-SD card and it must be formatted (FAT32) on Windows 10 environment.

Step 5: Python Program – SD Card Interface Module

MicroPython implements a Unix-like Virtual File System (VFS) layer. All mounted filesystems are combined into a single virtual filesystem, starting at the root /. Filesystems are mounted into directories in this structure, and at startup the working directory is changed to where the primary filesystem is mounted.

The main advantage of the FAT filesystem is that it can be accessed over USB MSC on supported boards without any additional drivers required on the host PC.

Download and store the MicroPython “sdcard.py” library into RPi Pico.

The file open functionality open(), can take following arguments for write, read and append a file.

w -- Open a file for writing the ASCII text. If a file is already existing, then it rewrites the file.

r -- Open a file for reading the ASCII text.

a -- Open a file for appending the ASCII text at the end.

wb -- Open a file for writing binary data.

rb -- Open a file for reading binary data.

MicroPython SPI, sdcard and os libraries are used for constructing the experiment.

MicroPython RPi Pico debug port is used for printing the debug statements.

During startup, the SPI bus and SD card is initialized.

Initially a “sample.txt” file is created / opened with FILE_WRITE option.

If file open is success, then the program writes running numbers (0 to19), into “sample.txt”, and then closes.

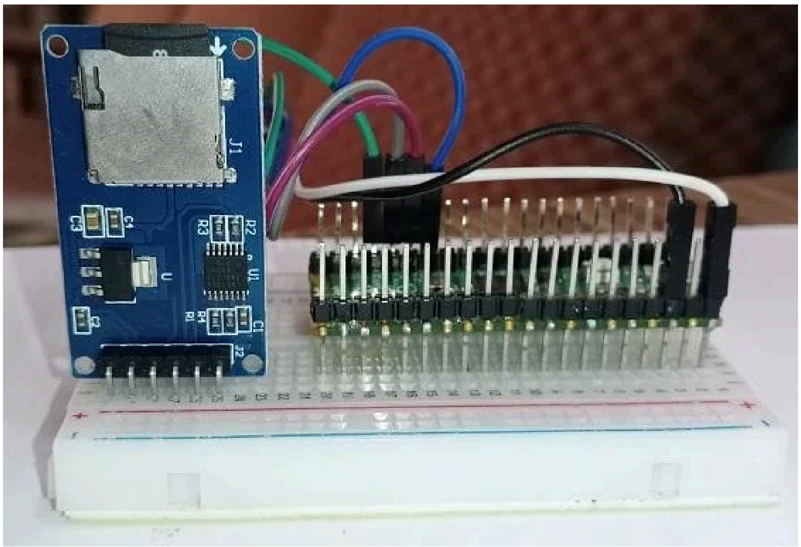
The program once again opens the “sample.txt” with FILE_APPEND option, and appends a text at the end.

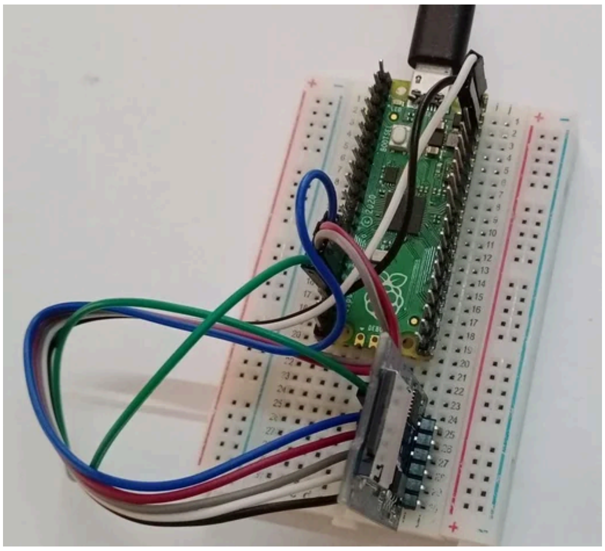
Then the program once again opens the “sample.txt” with FILE_READ option, reads the “sample.txt” and send data to the debug port, which prints text on connected debug terminal.

```
'''
 # Demonstrates RPi Pico interface to MicroSD Card Adapter
 # Create a text file and write running numbers.
 # Open text file, read and print the content on debug port

 * The Raspberry Pi Pico pin connections for MicroSD Card Adapter SPI

 # MicroSD Card Adapter Power Pins
 * MicroSD VCC pin to Pico VBUS
 * MicroSD  GND pin to Pico GND

 # MicroSD SPI Pins
 * MicroSD MISO pin to Pico GPIO-12
 * MicroSD MOSI pin to Pico GPIO-11
 * MicroSD SCK pin to Pico GPIO-10
 * MicroSD CS pin to Pico GPIO-13

 Name:- M.Pugazhendi
 Date:-  28thJul2021
 Version:- V0.1
 e-mail:- muthuswamy.pugazhendi@gmail.com
'''

from machine import Pin, SPI
import sdcard
import os

#Initialize the onboard LED as output
led = machine.Pin(25,machine.Pin.OUT)

# Toggle LED functionality
def BlinkLED(timer_one):
    led.toggle()

# Initialize the SD card
spi=SPI(1,baudrate=40000000,sck=Pin(10),mosi=Pin(11),miso=Pin(12))
sd=sdcard.SDCard(spi,Pin(13))

# Create a instance of MicroPython Unix-like Virtual File System (VFS),
vfs=os.VfsFat(sd)

# Mount the SD card
os.mount(sd,'/sd')

# Debug print SD card directory and files
print(os.listdir('/sd'))

# Create / Open a file in write mode.
# Write mode creates a new file.
# If  already file exists. Then, it overwrites the file.
file = open("/sd/sample.txt","w")

# Write sample text
for i in range(20):
    file.write("Sample text = %s\r\n" % i)

# Close the file
file.close()

# Again, open the file in "append mode" for appending a line
file = open("/sd/sample.txt","a")
file.write("Appended Sample Text at the END \n")
file.close()

# Open the file in "read mode".
# Read the file and print the text on debug port.
file = open("/sd/sample.txt", "r")
if file != 0:
    print("Reading from SD card")
    read_data = file.read()
    print (read_data)
file.close()

# Initialize timer_one. Used for toggling the on board LED
timer_one = machine.Timer()
```

```
# Timer one initialization for on board blinking LED at 200mS interval
timer_one.init(freq=5, mode=machine.Timer.PERIODIC, callback=BlinkLED)
```

Step 6: Conclusion

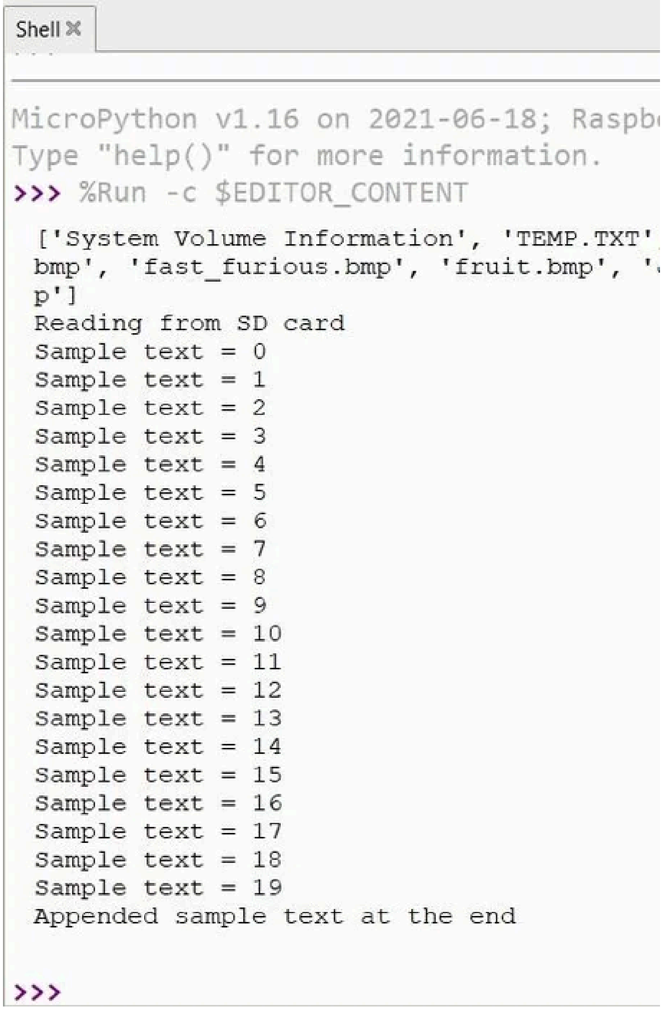




The experiment is successfully executed with RPi Pico, SD Card Module, and 8GB micro-SD card

A “sample.txt” file is created, written with running numbers and then readback / printed the text via debug port.

Step 7: Video Link



```
MicroPython v1.16 on 2021-06-18; Raspb
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT

 ['System Volume Information', 'TEMP.TXT'
 bmp', 'fast_furious.bmp', 'fruit.bmp', '
 p']
 Reading from SD card
 Sample text = 0
 Sample text = 1
 Sample text = 2
 Sample text = 3
 Sample text = 4
 Sample text = 5
 Sample text = 6
 Sample text = 7
 Sample text = 8
 Sample text = 9
 Sample text = 10
 Sample text = 11
 Sample text = 12
 Sample text = 13
 Sample text = 14
 Sample text = 15
 Sample text = 16
 Sample text = 17
 Sample text = 18
 Sample text = 19
 Appended sample text at the end

>>>
```

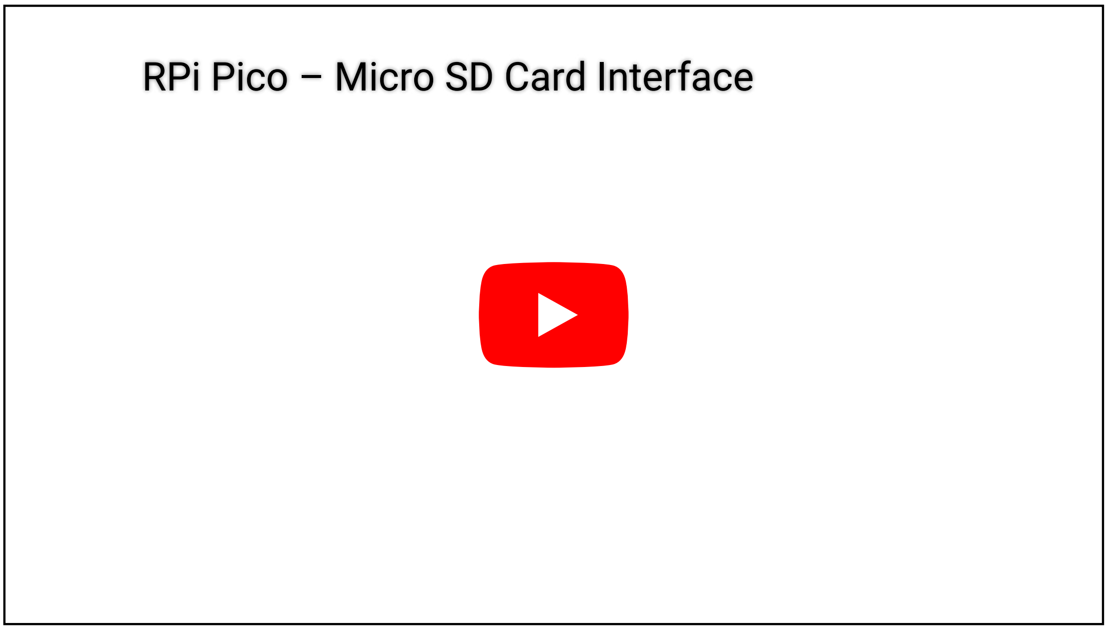
Result_Video.mp4

Visit "VESZLE - The Art Of Electronics" you tube channel for further information and videos.

https://www.youtube.com/channel/UCY4mekPLfeFinbQH...

If you enjoyed this instruct-able, then make sure you subscribe

The Micro SD card interface video link



RPi Pico – Micro SD Card Interface