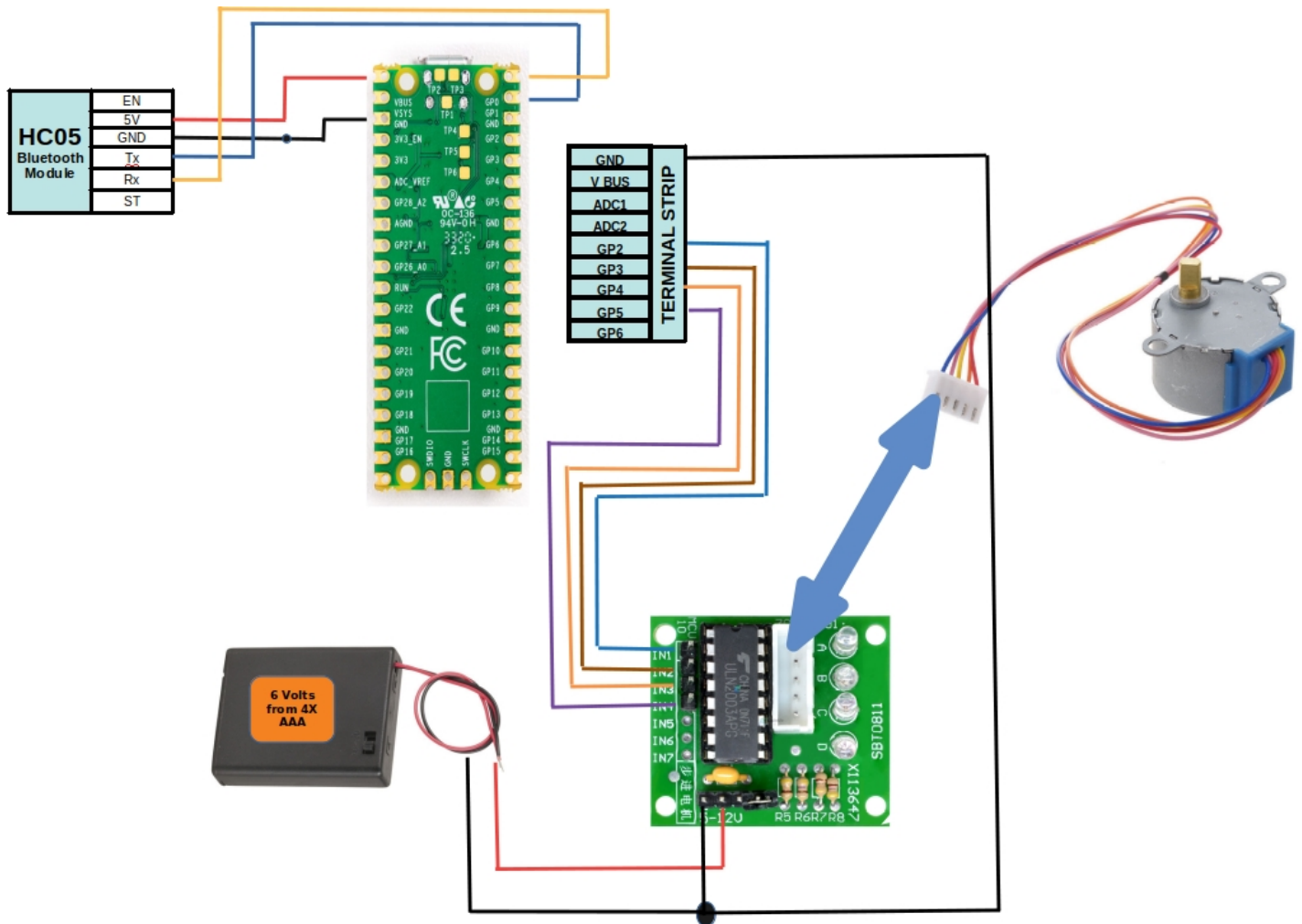


FINAL DESIGN OF FORWARD AND BACKWARD AND ROTATION CONTROL OF A STEPPER MOTOR VIA SERIAL TERMINAL ON AN ANDROID PHONE OVER BLUETOOTH.

THE CIRCUIT



THE COMPONENTS

Raspberry Pi Pico with Headers (from Core Electronics)
Raspberry Pi Pico socket eg : PiicoDev LiPo Expansion Board for Raspberry Pi Pico from Core Electronics
An HC05 Bluetooth Module (from Minikits in Adelaide)
Stepper Motor and Stepper Motor Controller (from Jaycar)
Solid core hookup wire
Short length of terminal strip
Android Phone Bluetooth Terminal App : <http://www.kai-morich.de/android/>

THE MICROPYTHON PROGRAM SCRIPT

(Note it is essential that the leading tabs are strictly observed otherwise the script will not run.)

```
from machine import Pin
import utime

#Bluetooth coding

from machine import Pin,UART
uart = UART(0,9600)

pins = [
    Pin(2,Pin.OUT),
    Pin(3,Pin.OUT),
    Pin(4,Pin.OUT),
    Pin(5,Pin.OUT),
    ]

full_step_sequence1 = [
    [0,0,0,1],
    [0,0,1,0],
    [0,1,0,0],
    [1,0,0,0]
    ]

full_step_sequence2 = [
    [1,0,0,0],
    [0,1,0,0],
    [0,0,1,0],
    [0,0,0,1]
    ]

# print('checking BT')
# print('1 = clockwise, 2 = anticlockwise')
# direction = input("Direction code = ")

while True:

# Ask user for direction of rotation ...
    uart.write('1 = clockwise, 2 = anticlockwise \n')
    direction = uart.readline()
    print(direction)
```

```

while uart.any() == 0:
    #print('Waiting for user input')
    direction = 'Unknown'
    if uart.any() > 0 :
        break

direction = uart.readline()

print(direction)

direction = int(direction)

print('Converted HC05 input ... ', direction)

# Ask user for number of revolutions ...
uart.write('How many revolutions ? \n')
uart.flush()
revs = uart.readline()
print(revs)

while uart.any() == 0:
    #print('Waiting for user input')
    revs = 'Unknown'
    if uart.any() > 0 :
        break

revs = uart.readline()
print(revs)
revs = int(revs)
print('Converted HC05 input REVS parameter... ', revs)
revs = revs * 512

if direction == 2:
    print('Direction = ANTI CLOCKWISE')
    for j in range(revs):
        for step in full_step_sequence2:
            for i in range(len(pins)):
                pins[i].value(step[i])
                utime.sleep(0.001)

if direction == 1:

```

```

print('Direction = CLOCKWISE')
for j in range(revs):
    for step in full_step_sequence1:
        for i in range(len(pins)):
            pins[i].value(step[i])
            utime.sleep(0.001)

#Turn power off to the windings .... Set all to 0
print('Turning 6V power supply to off on each winding .... Set all to 0')
for i in range(4):
    j = i+1
    print('Pin', j, " OFF")
    pins[i].value(0)

```

TYPICAL DIALOG AS VIEWED ON THE THONNY IDE

```

>>> %Run -c $EDITOR_CONTENT
None
b'2'
Converted HC05 input ... 2
b'\r\n'
b'2'
Converted HC05 input REVS parameter... 2
Direction = ANTI CLOCKWISE
Turning 6V power supply to off on each winding .... Set all to 0
Pin 1 OFF
Pin 2 OFF
Pin 3 OFF
Pin 4 OFF
b'\r\n'
b'1'
Converted HC05 input ... 1
b'\r\n'
b'1'
Converted HC05 input REVS parameter... 1
Direction = CLOCKWISE
Turning 6V power supply to off on each winding .... Set all to 0
Pin 1 OFF
Pin 2 OFF
Pin 3 OFF
Pin 4 OFF

```

```
b'\r\n'  
b'2'  
Converted HC05 input ... 2  
b'\r\n'  
b'2'  
Converted HC05 input REVS parameter... 2  
Direction = ANTI CLOCKWISE  
Turning 6V power supply to off on each winding .... Set all to 0  
Pin 1 OFF  
Pin 2 OFF  
Pin 3 OFF  
Pin 4 OFF  
b'\r\n'
```

TYPICAL DIALOG AS VIEWED ON THE BLUETOOTH SERIAL TERMINAL APP ON AN ANDROID PHONE



Queries and further information available from tomhartley850@gmail.com