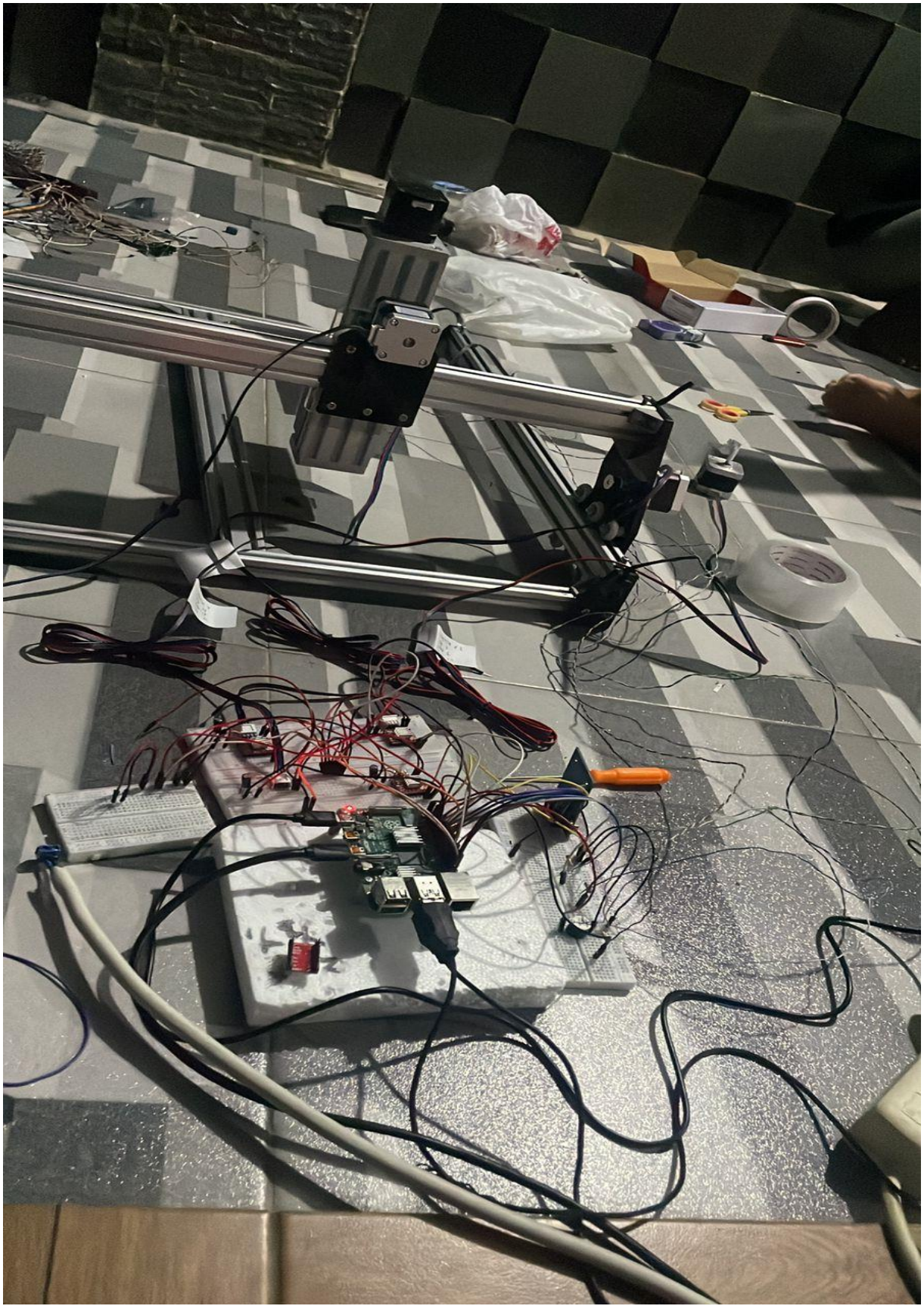
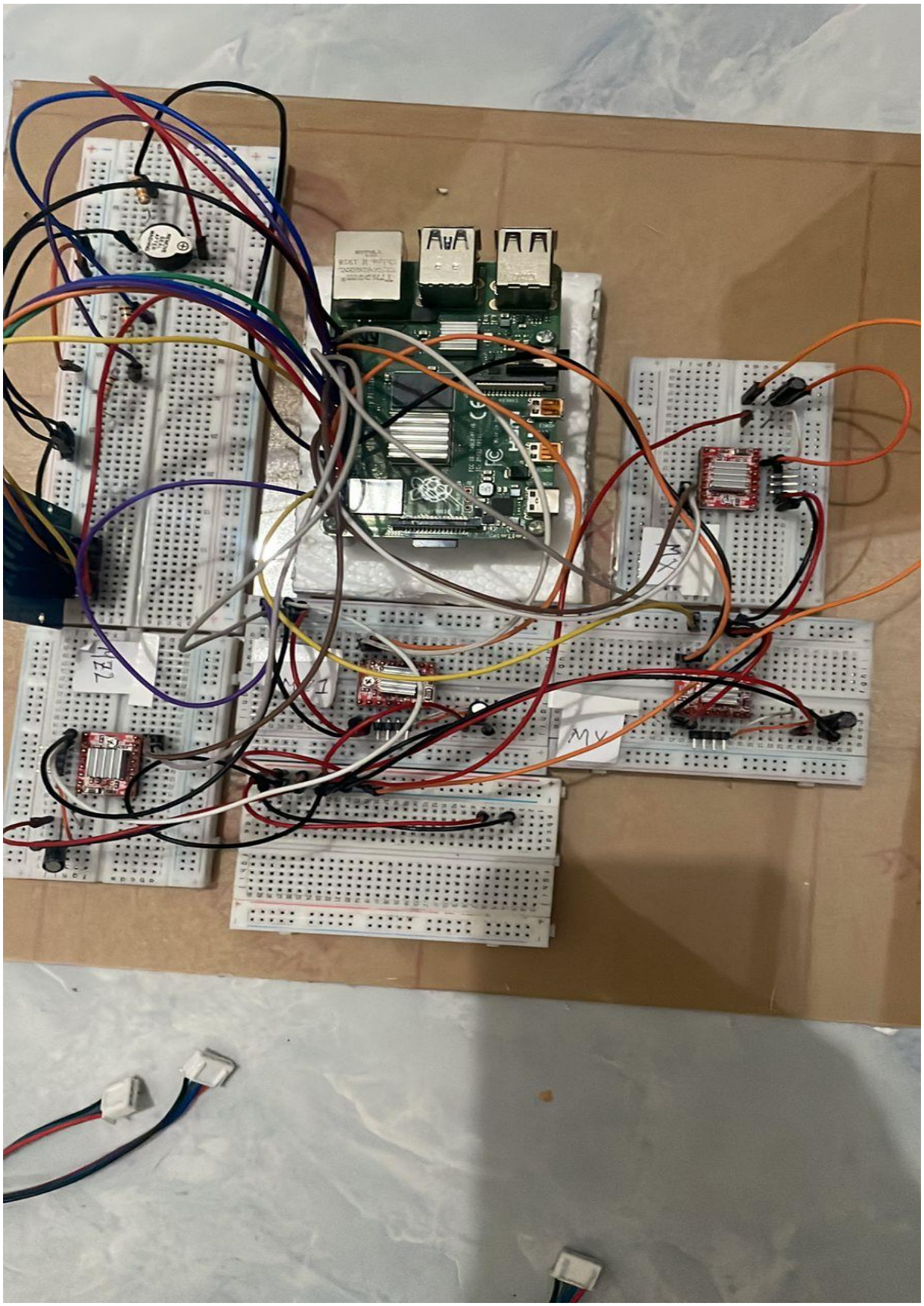
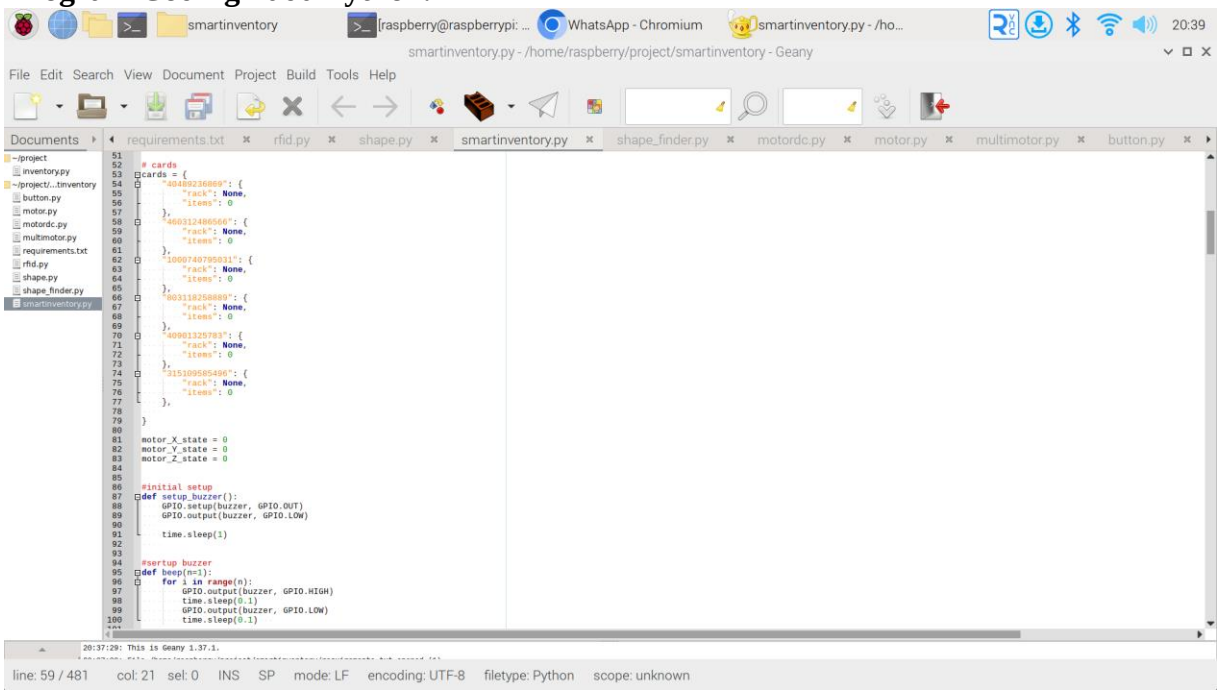
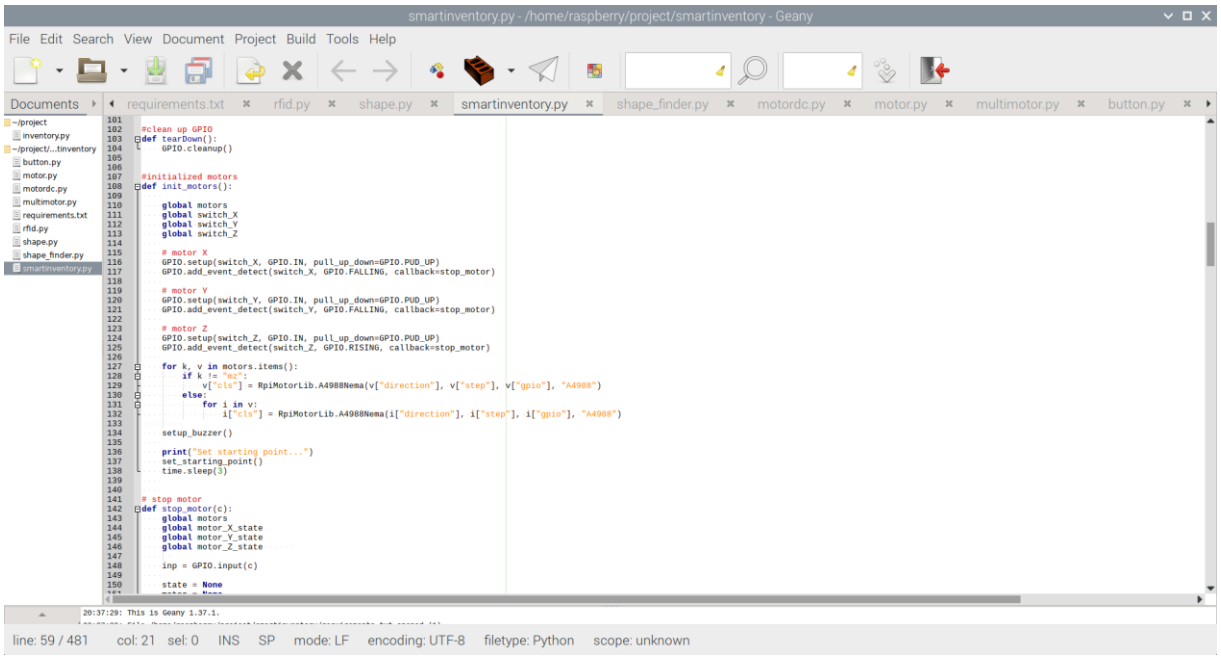
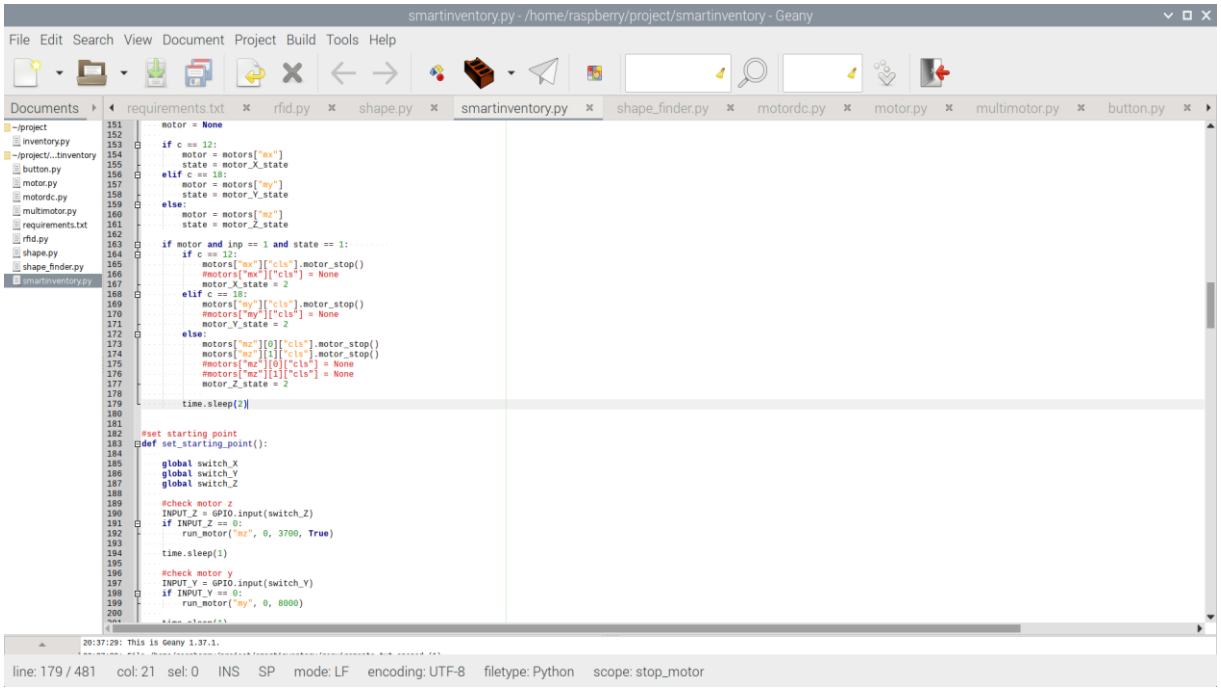


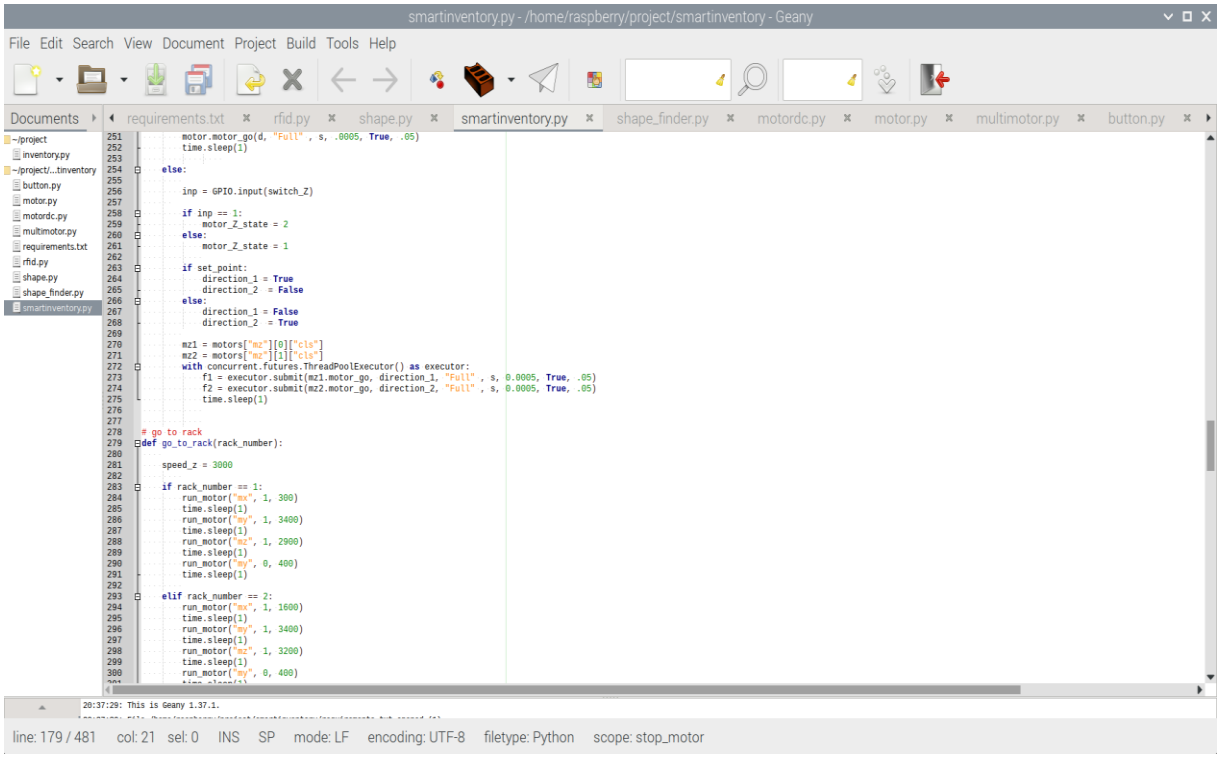


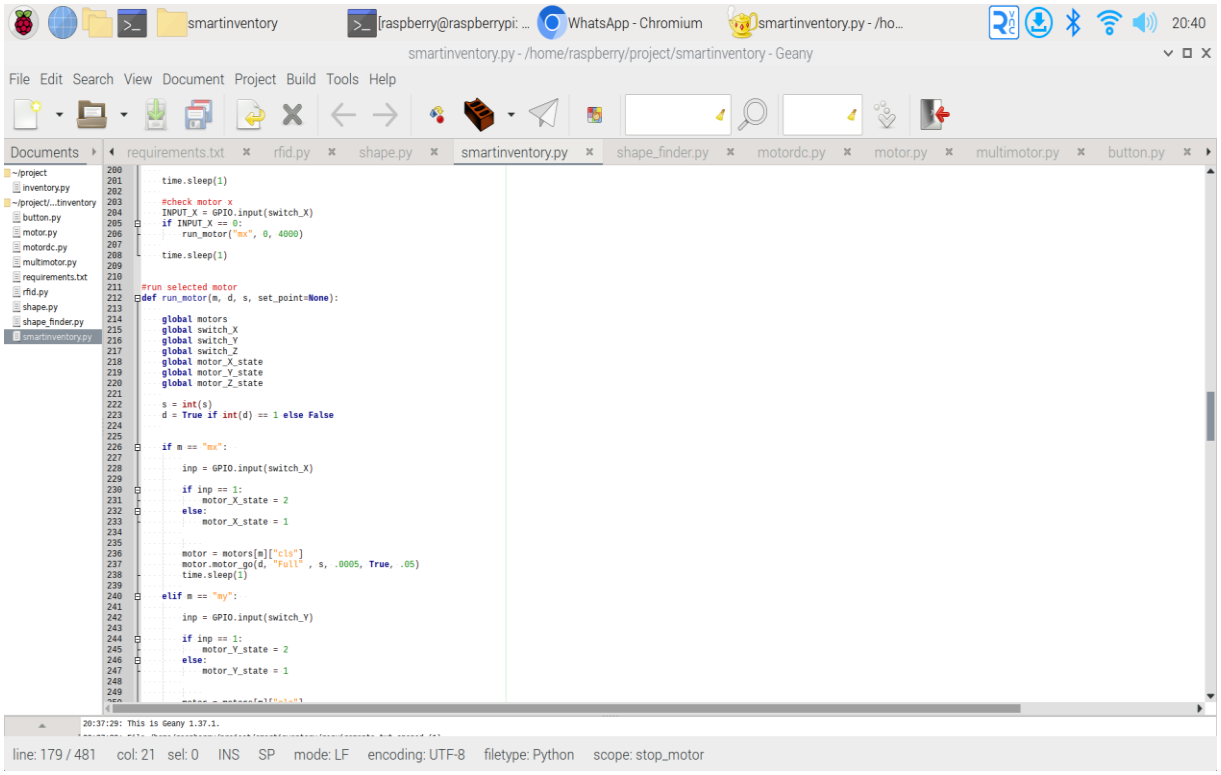
Program Coding Pada Python.

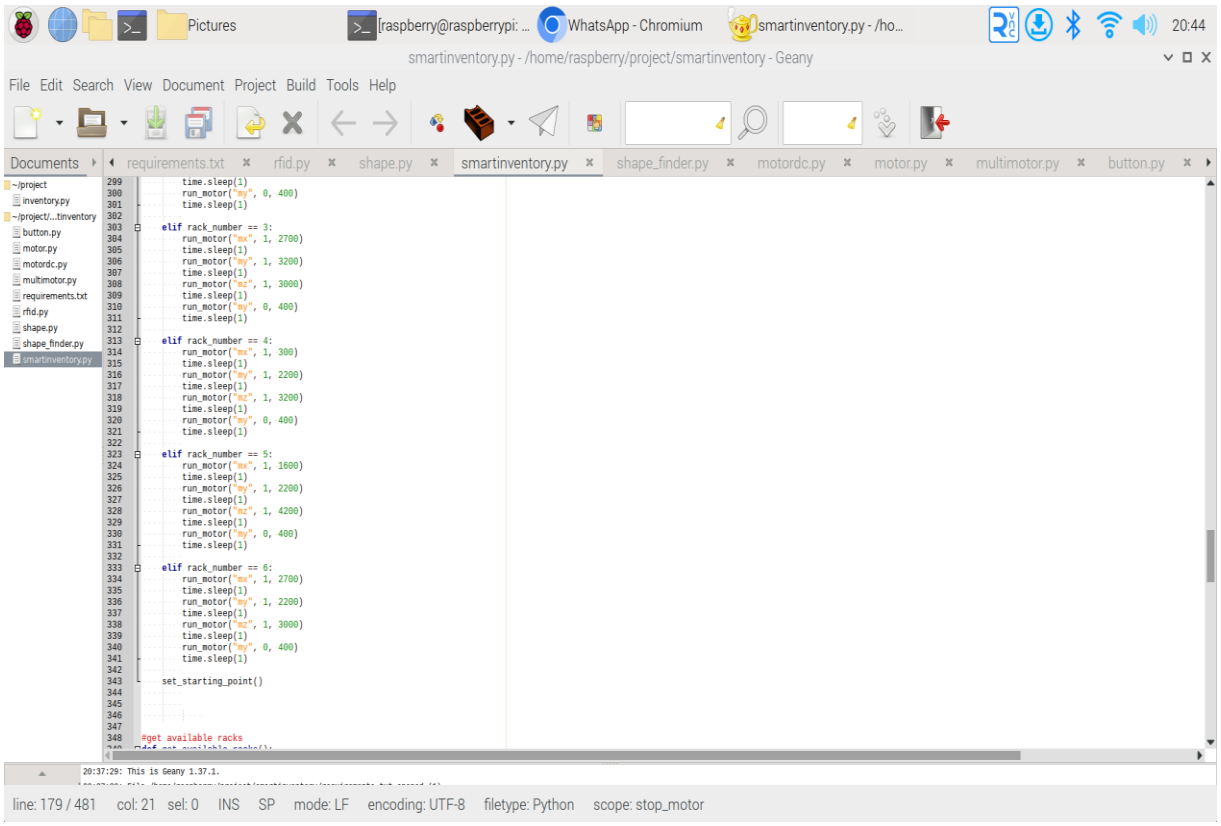


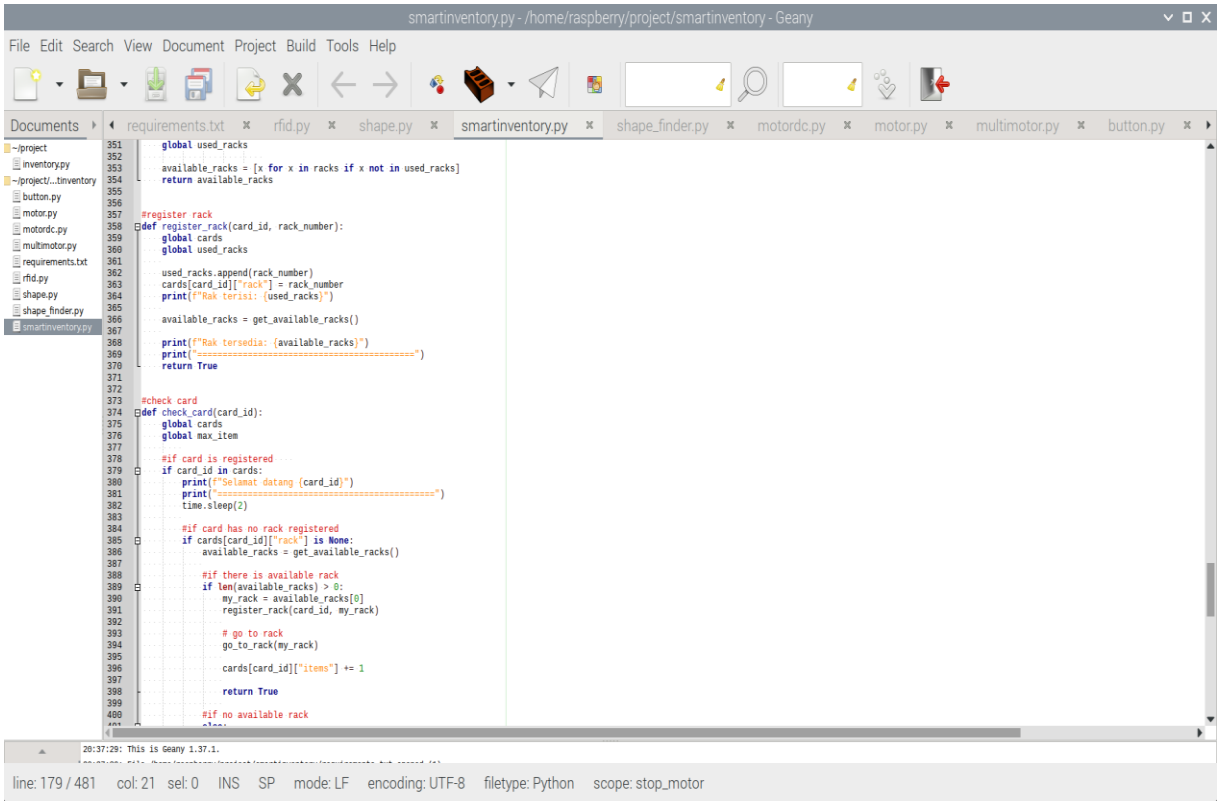


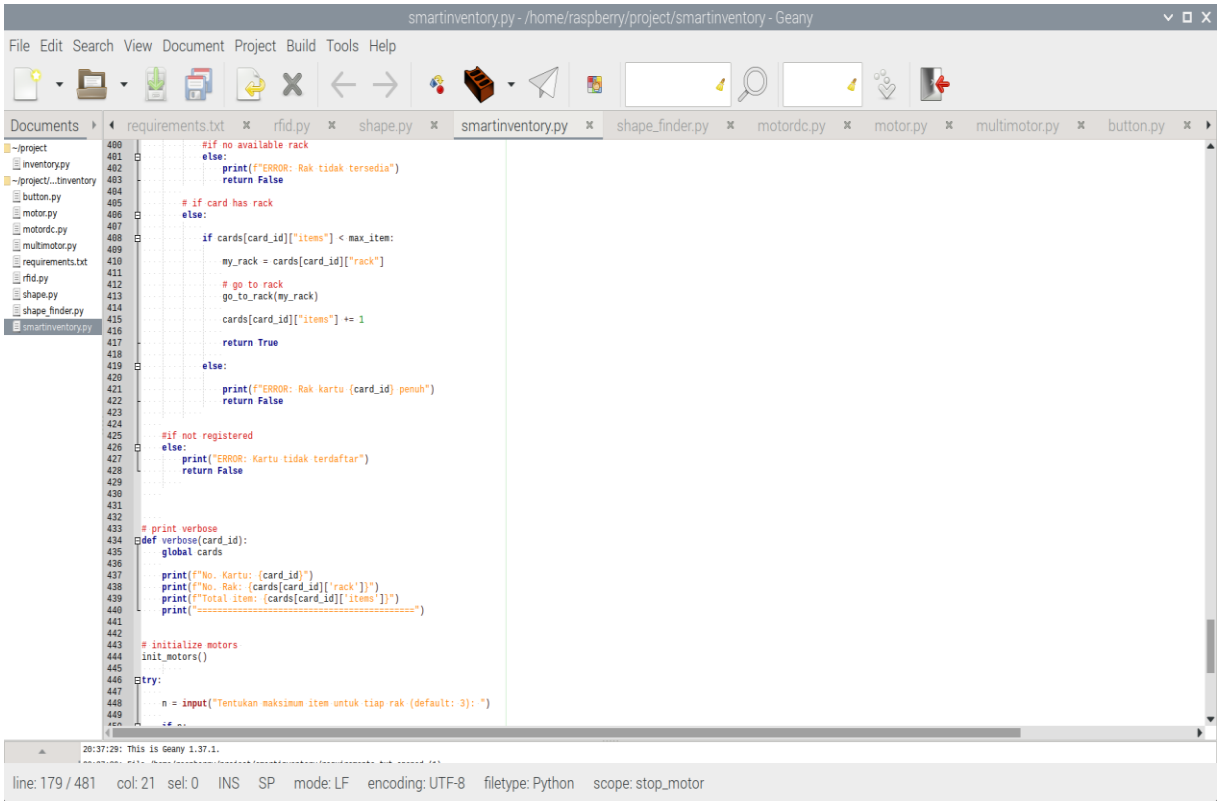

smartinventory.py - /home/raspberry/project/smartinventory - Geany
File Edit Search View Document Project Build Tools Help
Documents
requirements.txt
rfid.py
shape.py
smartinventory.py
shape_finder.py
motordc.py
motor.py
multimotor.py
button.py
motor = None
if c == 12:
motor = motors["mx"]
state = motor_X_state
elif c == 18:
motor = motors["my"]
state = motor_Y_state
else:
motor = motors["mz"]
state = motor_Z_state
if motor and inp == 1 and state == 1:
if c == 12:
motors["mx"]["cls"].motor_stop()
#motors["mx"]["cls"] = None
motor_X_state = 2
elif c == 18:
motors["my"]["cls"].motor_stop()
#motors["my"]["cls"] = None
motor_Y_state = 2
else:
motors["mz"][0]["cls"].motor_stop()
motors["mz"][1]["cls"].motor_stop()
#motors["mz"][0]["cls"] = None
#motors["mz"][1]["cls"] = None
motor_Z_state = 2
time.sleep(2)
#set starting point
def set_starting_point():
global switch_X
global switch_Y
global switch_Z
#check motor z
INPUT_Z = GPIO.input(switch_Z)
if INPUT_Z == 0:
run_motor("mz", 0, 3700, True)
time.sleep(1)
#check motor y
INPUT_Y = GPIO.input(switch_Y)
if INPUT_Y == 0:
run_motor("my", 0, 8000)
20:37:29: This is Geany 1.37.1.
line: 179 / 481 col: 21 sel: 0 INS SP mode: LF encoding: UTF-8 filetype: Python scope: stop_motor

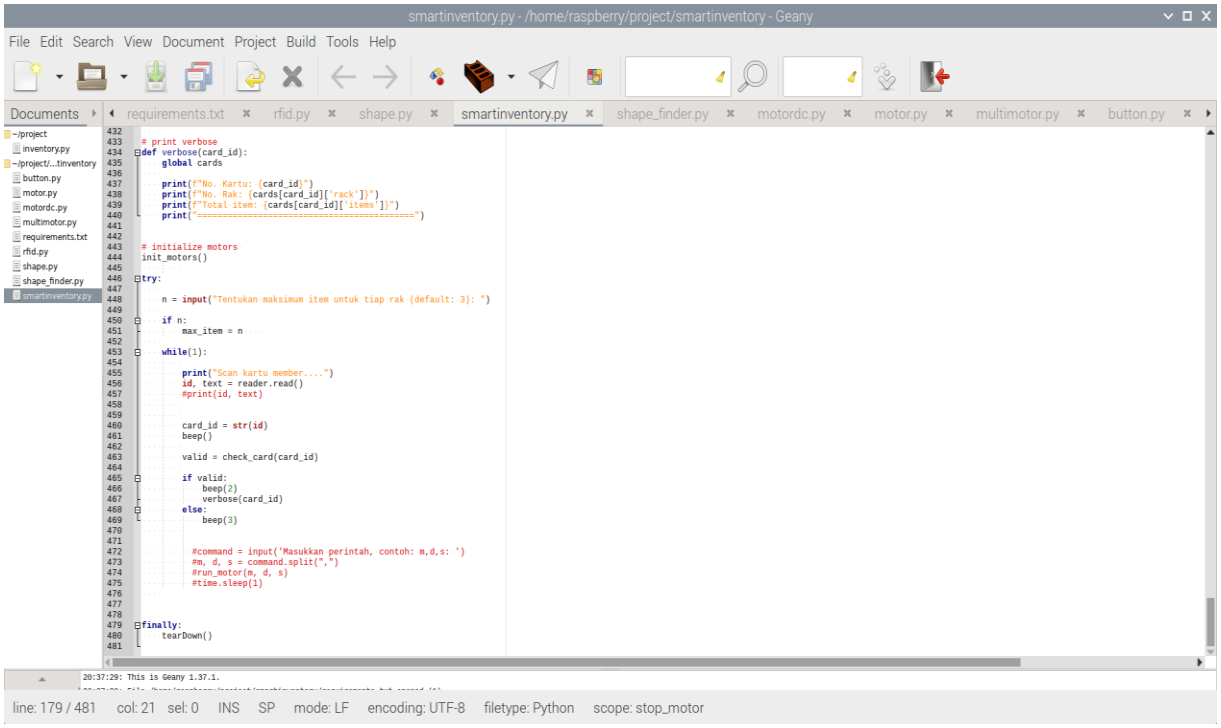

smartinventory.py - /home/raspberry/project/smartinventory - Geany
File Edit Search View Document Project Build Tools Help
Documents
requirements.txt
rfid.py
shape.py
smartinventory.py
shape_finder.py
motordc.py
motor.py
multimotor.py
button.py
motor.motor_go(d, "Full" , s, .0005, True, .05)
time.sleep(1)
else:
inp = GPIO.input(switch_Z)
if inp == 1:
motor_Z_state = 2
else:
motor_Z_state = 1
if set_point:
direction_1 = True
direction_2 = False
else:
direction_1 = False
direction_2 = True
mz1 = motors["mz"][0]["cls"]
mz2 = motors["mz"][1]["cls"]
with concurrent.futures.ThreadPoolExecutor() as executor:
f1 = executor.submit(mz1.motor_go, direction_1, "Full" , s, 0.0005, True, .05)
f2 = executor.submit(mz2.motor_go, direction_2, "Full" , s, 0.0005, True, .05)
time.sleep(1)
# go to rack
def go_to_rack(rack_number):
speed_z = 3000
if rack_number == 1:
run_motor("mx", 1, 300)
time.sleep(1)
run_motor("my", 1, 3400)
time.sleep(1)
run_motor("mz", 1, 2900)
time.sleep(1)
run_motor("my", 0, 400)
time.sleep(1)
elif rack_number == 2:
run_motor("mx", 1, 1600)
time.sleep(1)
run_motor("my", 1, 3400)
time.sleep(1)
run_motor("mz", 1, 3200)
time.sleep(1)
run_motor("my", 0, 400)
20:37:29: This is Geany 1.37.1.
line: 179 / 481 col: 21 sel: 0 INS SP mode: LF encoding: UTF-8 filetype: Python scope: stop_motor


smartinventory
[raspberry@raspberrypi: ...
WhatsApp - Chromium
smartinventory.py - /ho...
20:40
smartinventory.py - /home/raspberry/project/smartinventory - Geany
File Edit Search View Document Project Build Tools Help
Documents
requirements.txt
rfid.py
shape.py
smartinventory.py
shape_finder.py
motordc.py
motor.py
multimotor.py
button.py
~/project
inventory.py
~/project/...tinventory
button.py
motor.py
motordc.py
multimotor.py
requirements.txt
rfid.py
shape.py
shape_finder.py
smartinventory.py
time.sleep(1)
#check motor x
INPUT_X = GPIO.input(switch_X)
if INPUT_X == 0:
run_motor("mx", 0, 4000)
time.sleep(1)
#run selected motor
def run_motor(m, d, s, set_point=None):
global motors
global switch_X
global switch_Y
global switch_Z
global motor_X_state
global motor_Y_state
global motor_Z_state
s = int(s)
d = True if int(d) == 1 else False
if m == "mx":
inp = GPIO.input(switch_X)
if inp == 1:
motor_X_state = 2
else:
motor_X_state = 1
motor = motors[m]["cls"]
motor.motor_go(d, "Full" , s, .0005, True, .05)
time.sleep(1)
elif m == "my":
inp = GPIO.input(switch_Y)
if inp == 1:
motor_Y_state = 2
else:
motor_Y_state = 1
20:37:29: This is Geany 1.37.1.
line: 179 / 481 col: 21 sel: 0 INS SP mode: LF encoding: UTF-8 filetype: Python scope: stop_motor

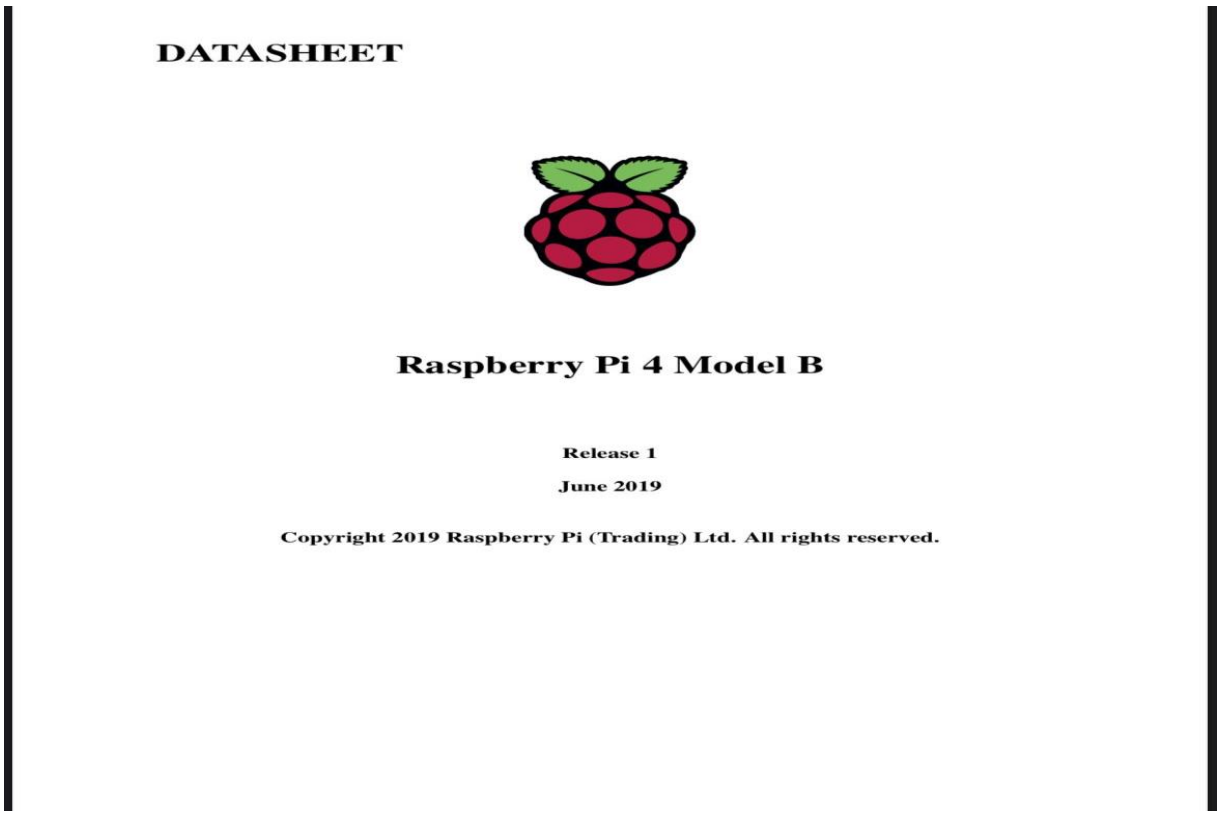

Pictures
[raspberry@raspberrypi: ...
WhatsApp - Chromium
smartinventory.py - /ho...
20:44
smartinventory.py - /home/raspberry/project/smartinventory - Geany
File Edit Search View Document Project Build Tools Help
Documents
requirements.txt
rfid.py
shape.py
smartinventory.py
shape_finder.py
motordc.py
motor.py
multimotor.py
button.py
time.sleep(1)
run_motor("my", 0, 400)
time.sleep(1)
elif rack_number == 3:
run_motor("mx", 1, 2700)
time.sleep(1)
run_motor("my", 1, 3200)
time.sleep(1)
run_motor("mz", 1, 3000)
time.sleep(1)
run_motor("my", 0, 400)
time.sleep(1)
elif rack_number == 4:
run_motor("mx", 1, 300)
time.sleep(1)
run_motor("my", 1, 2200)
time.sleep(1)
run_motor("mz", 1, 3200)
time.sleep(1)
run_motor("my", 0, 400)
time.sleep(1)
elif rack_number == 5:
run_motor("mx", 1, 1600)
time.sleep(1)
run_motor("my", 1, 2200)
time.sleep(1)
run_motor("mz", 1, 4200)
time.sleep(1)
run_motor("my", 0, 400)
time.sleep(1)
elif rack_number == 6:
run_motor("mx", 1, 2700)
time.sleep(1)
run_motor("my", 1, 2200)
time.sleep(1)
run_motor("mz", 1, 3000)
time.sleep(1)
run_motor("my", 0, 400)
time.sleep(1)
set_starting_point()
#get available racks
20:37:29: This is Geany 1.37.1.
line: 179 / 481 col: 21 sel: 0 INS SP mode: LF encoding: UTF-8 filetype: Python scope: stop_motor


smartinventory.py - /home/raspberry/project/smartinventory - Geany
File Edit Search View Document Project Build Tools Help
Documents
requirements.txt
rfid.py
shape.py
smartinventory.py
shape_finder.py
motordc.py
motor.py
multimotor.py
button.py
~/project
inventory.py
~/project/...tinventory
button.py
motor.py
motordc.py
multimotor.py
requirements.txt
rfid.py
shape.py
shape_finder.py
smartinventory.py
global used_racks
available_racks = [x for x in racks if x not in used_racks]
return available_racks
#register rack
def register_rack(card_id, rack_number):
global cards
global used_racks
used_racks.append(rack_number)
cards[card_id]["rack"] = rack_number
print(f"Rak terisi: {used_racks}")
available_racks = get_available_racks()
print(f"Rak tersedia: {available_racks}")
print("==")
return True
#check card
def check_card(card_id):
global cards
global max_item
#if card is registered
if card_id in cards:
print(f"Selamat datang {card_id}")
print("==")
time.sleep(2)
#if card has no rack registered
if cards[card_id]["rack"] is None:
available_racks = get_available_racks()
#if there is available rack
if len(available_racks) > 0:
my_rack = available_racks[0]
register_rack(card_id, my_rack)
# go to rack
go_to_rack(my_rack)
cards[card_id]["items"] += 1
return True
#if no available rack
20:37:29: This is Geany 1.37.1.
line: 179 / 481 col: 21 sel: 0 INS SP mode: LF encoding: UTF-8 filetype: Python scope: stop_motor


smartinventory.py - /home/raspberry/project/smartinventory - Geany
File Edit Search View Document Project Build Tools Help
Documents
requirements.txt
rfid.py
shape.py
smartinventory.py
shape_finder.py
motordc.py
motor.py
multimotor.py
button.py
#if no available rack
else:
print(f"ERROR: Rak tidak tersedia")
return False
# if card has rack
else:
if cards[card_id]["items"] < max_item:
my_rack = cards[card_id]["rack"]
# go to rack
go_to_rack(my_rack)
cards[card_id]["items"] += 1
return True
else:
print(f"ERROR: Rak kartu {card_id} penuh")
return False
#if not registered
else:
print("ERROR: Kartu tidak terdaftar")
return False
# print verbose
def verbose(card_id):
global cards
print(f"No. Kartu: {card_id}")
print(f"No. Rak: {cards[card_id]['rack']}")
print(f"Total item: {cards[card_id]['items']}")
print("==")
# initialize motors
init_motors()
try:
n = input("Tentukan maksimum item untuk tiap rak (default: 3): ")
20:37:29: This is Geany 1.37.1.
line: 179 / 481 col: 21 sel: 0 INS SP mode: LF encoding: UTF-8 filetype: Python scope: stop_motor



Data Sheet Raspberry Pi 3



DATASHEET

Raspberry Pi 4 Model B

Release 1

June 2019

2 Features

2.1 Hardware

- Quad core 64-bit ARM-Cortex A72 running at 1.5GHz
- 1, 2 and 4 Gigabyte LPDDR4 RAM options
- H.265 (HEVC) hardware decode (up to 4Kp60)
- H.264 hardware decode (up to 1080p60)
- VideoCore VI 3D Graphics
- Supports dual HDMI display output up to 4Kp60

2.2 Interfaces

- 802.11 b/g/n/ac Wireless LAN
- Bluetooth 5.0 with BLE
- 1x SD Card
- 2x micro-HDMI ports supporting dual displays up to 4Kp60 resolution
- 2x USB2 ports
- 2x USB3 ports
- 1x Gigabit Ethernet port (supports PoE with add-on PoE HAT)
- 1x Raspberry Pi camera port (2-lane MIPI CSI)
- 1x Raspberry Pi display port (2-lane MIPI DSI)
- 28x user GPIO supporting various interface options:
  - Up to 6x UART
  - Up to 6x I2C
  - Up to 5x SPI
  - 1x SDIO interface
  - 1x DPI (Parallel RGB Display)
  - 1x PCM
  - Up to 2x PWM channels
  - Up to 3x GPCLK outputs

2.3 Software

- ARMv8 Instruction Set
- Mature Linux software stack
- Actively developed and maintained
  - Recent Linux kernel support
  - Many drivers upstreamed
  - Stable and well supported userland
  - Availability of GPU functions using standard APIs

3 Mechanical Specification

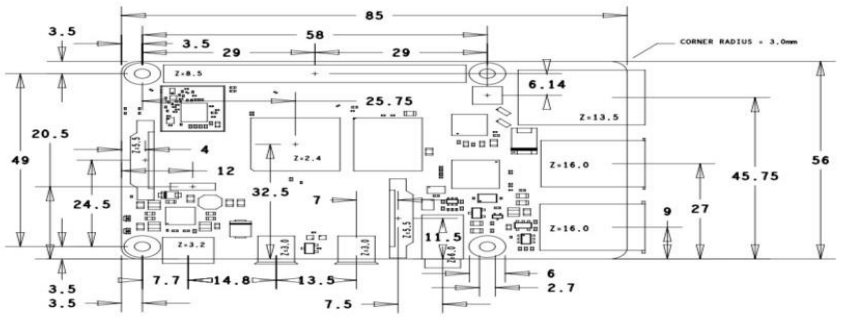


Figure 1: Mechanical Dimensions

4 Electrical Specification

Caution! Stresses above those listed in Table 2 may cause permanent damage to the device. This is a stress rating only; functional operation of the device under these or any other conditions above those listed in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Symbol	Parameter	Minimum	Maximum	Unit
VIN	5V Input Voltage	-0.5	6.0	V

Table 2: Absolute Maximum Ratings

Please note that VDD_IO is the GPIO bank voltage which is tied to the on-board 3.3V supply rail.

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{IL}	Input low voltage[a]	VDD_IO = 3.3V	-	-	TBD	V
V_{IH}	Input high voltage[a]	VDD_IO = 3.3V	TBD	-	-	V
I_{IL}	Input leakage current	TA = +85°C	-	-	TBD	μA
C_{IN}	Input capacitance	-	-	TBD	-	pF
V_{OL}	Output low voltage[b]	VDD_IO = 3.3V, IOL = -2mA	-	-	TBD	V
V_{OH}	Output high voltage[b]	VDD_IO = 3.3V, IOH = 2mA	TBD	-	-	V
I_{OL}	Output low current[c]	VDD_IO = 3.3V, VO = 0.4V	TBD	-	-	mA
I_{OH}	Output high current[c]	VDD_IO = 3.3V, VO = 2.3V	TBD	-	-	mA
R_{PU}	Pullup resistor	-	TBD	-	TBD	kΩ
R_{PD}	Pulldown resistor	-	TBD	-	TBD	kΩ

[a] Hysteresis enabled
[b] Default drive strength (8mA)
[c] Maximum drive strength (16mA)

Table 3: DC Characteristics

Pin Name	Symbol	Parameter	Minimum	Typical	Maximum	Unit
Digital outputs	t_{rise}	10-90% rise time[a]	-	TBD	-	ns
Digital outputs	t_{fall}	90-10% fall time[a]	-	TBD	-	ns

[a] Default drive strength, CL = 5pF, VDD_IO = 3.3V

Table 4: Digital I/O Pin AC Characteristics

Figure 2: Digital IO Characteristics

4.1 Power Requirements

The Pi4B requires a good quality USB-C power supply capable of delivering 5V at 3A. If attached downstream USB devices consume less than 500mA, a 5V, 2.5A supply may be used.

5 Peripherals

5.1 GPIO Interface

The Pi4B makes 28 BCM2711 GPIOs available via a standard Raspberry Pi 40-pin header. This header is backwards compatible with all previous Raspberry Pi boards with a 40-way header.

5.1.1 GPIO Pin Assignments

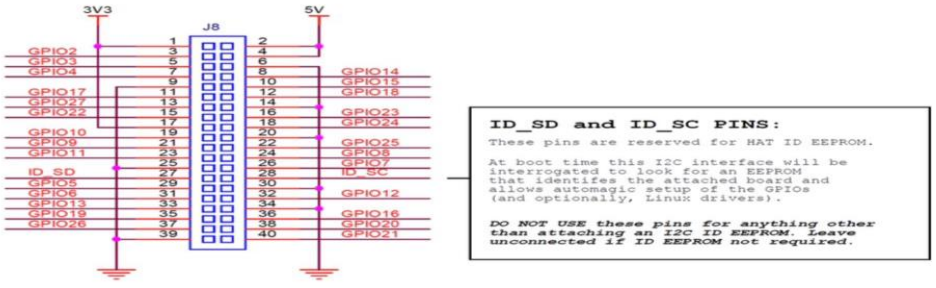


Figure 3: GPIO Connector Pinout

As well as being able to be used as straightforward software controlled input and output (with programmable pulls), GPIO pins can be switched (multiplexed) into various other modes backed by dedicated peripheral blocks such as I2C, UART and SPI.

In addition to the standard peripheral options found on legacy Pis, extra I2C, UART and SPI peripherals have been added to the BCM2711 chip and are available as further mux options on the Pi4. This gives users much more flexibility when attaching add-on hardware as compared to older models.

5.1.2 GPIO Alternate Functions

GPIO	Default Pull	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5
0	High	SDA0	SA5	PCLK	SPI3_CE0_N	TXD2	SDA6
1	High	SCL0	SA4	DE	SPI3_MISO	RXD2	SCL6
2	High	SDA1	SA3	LCD_VSYNC	SPI3_MOSI	CTS2	SDA3
3	High	SCL1	SA2	LCD_HSYNC	SPI3_SCLK	RTS2	SCL3
4	High	GPCLK0	SA1	DPI_D0	SPI4_CE0_N	TXD3	SDA3
5	High	GPCLK1	SA0	DPI_D1	SPI4_MISO	RXD3	SCL3
6	High	GPCLK2	SOE_N	DPI_D2	SPI4_MOSI	CTS3	SDA4
7	High	SPI0_CE1_N	SWE_N	DPI_D3	SPI4_SCLK	RTS3	SCL4
8	High	SPI0_CE0_N	SD0	DPI_D4	-	TXD4	SDA4
9	Low	SPI0_MISO	SD1	DPI_D5	-	RXD4	SCL4
10	Low	SPI0_MOSI	SD2	DPI_D6	-	CTS4	SDA5
11	Low	SPI0_SCLK	SD3	DPI_D7	-	RTS4	SCL5
12	Low	PWM0	SD4	DPI_D8	SPI5_CE0_N	TXD5	SDA5
13	Low	PWM1	SD5	DPI_D9	SPI5_MISO	RXD5	SCL5
14	Low	TXD0	SD6	DPI_D10	SPI5_MOSI	CTS5	TXD1
15	Low	RXD0	SD7	DPI_D11	SPI5_SCLK	RTS5	RXD1
16	Low	FL0	SD8	DPI_D12	CTS0	SPI1_CE2_N	CTS1
17	Low	FL1	SD9	DPI_D13	RTS0	SPI1_CE1_N	RTS1
18	Low	PCM_CLK	SD10	DPI_D14	SPI6_CE0_N	SPI1_CE0_N	PWM0
19	Low	PCM_FS	SD11	DPI_D15	SPI6_MISO	SPI1_MISO	PWM1
20	Low	PCM_DIN	SD12	DPI_D16	SPI6_MOSI	SPI1_MOSI	GPCLK0
21	Low	PCM_DOUT	SD13	DPI_D17	SPI6_SCLK	SPI1_SCLK	GPCLK1
22	Low	SD0_CLK	SD14	DPI_D18	SD1_CLK	ARM_TRST	SDA6
23	Low	SD0_CMD	SD15	DPI_D19	SD1_CMD	ARM_RTCK	SCL6
24	Low	SD0_DAT0	SD16	DPI_D20	SD1_DAT0	ARM_TDO	SPI3_CE1_N
25	Low	SD0_DAT1	SD17	DPI_D21	SD1_DAT1	ARM_TCK	SPI4_CE1_N
26	Low	SD0_DAT2	TE0	DPI_D22	SD1_DAT2	ARM_TDI	SPI5_CE1_N
27	Low	SD0_DAT3	TE1	DPI_D23	SD1_DAT3	ARM_TMS	SPI6_CE1_N

Table 5: Raspberry Pi 4 GPIO Alternate Functions

Table 5 details the default pin pull state and available alternate GPIO functions. Most of these alternate peripheral functions are described in detail in the BCM2711 Peripherals Specification document which can be downloaded from the hardware documentation section of the website.

5.1.3 Display Parallel Interface (DPI)

A standard parallel RGB (DPI) interface is available the GPIOs. This up-to-24-bit parallel interface can support a secondary display.

5.1.4 SD/SDIO Interface

The Pi4B has a dedicated SD card socket which suports 1.8V, DDR50 mode (at a peak bandwidth of 50 Megabytes / sec). In addition, a legacy SDIO interface is available on the GPIO pins.

5.2 Camera and Display Interfaces

The Pi4B has 1x Raspberry Pi 2-lane MIPI CSI Camera and 1x Raspberry Pi 2-lane MIPI DSI Display connector. These connectors are backwards compatible with legacy Raspberry Pi boards, and support all of the available Raspberry Pi camera and display peripherals.

5.3 USB

The Pi4B has 2x USB2 and 2x USB3 type-A sockets. Downstream USB current is limited to approximately 1.1A in aggregate over the four sockets.

5.4 HDMI

The Pi4B has 2x micro-HDMI ports, both of which support CEC and HDMI 2.0 with resolutions up to 4Kp60.

5.5 Audio and Composite (TV Out)

The Pi4B supports near-CD-quality analogue audio output and composite TV-output via a 4-ring TRS 'A/V' jack.

The analog audio output can drive 32 Ohm headphones directly.

5.6 Temperature Range and Thermals

The recommended ambient operating temperature range is 0 to 50 degrees Celcius.

To reduce thermal output when idling or under light load, the Pi4B reduces the CPU clock speed and voltage. During heavier load the speed and voltage (and hence thermal output) are increased. The internal governor will throttle back both the CPU speed and voltage to make sure the CPU temperature never exceeds 85 degrees C.

The Pi4B will operate perfectly well without any extra cooling and is designed for sprint performance - expecting a light use case on average and ramping up the CPU speed when needed (e.g. when loading a webpage). If a user wishes to load the system continually or operate it at a high termperature at full performance, further cooling may be needed.

Data *Sheet Driver Motor* A4988

A4988

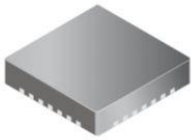
DMOS Microstepping Driver with Translator and Overcurrent Protection

Features and Benefits

- Low $R_{DS(ON)}$ outputs
- Automatic current decay mode detection/selection
- Mixed and Slow current decay modes
- Synchronous rectification for low power dissipation
- Internal UVLO
- Crossover-current protection
- 3.3 and 5 V compatible logic supply
- Thermal shutdown circuitry
- Short-to-ground protection
- Shorted load protection
- Five selectable step modes: full, $^1/_2$, $^1/_4$, $^1/_8$, and $^1/_{16}$

Package:

28-contact QFN
with exposed thermal pad
5 mm × 5 mm × 0.90 mm
(ET package)



Approximate size

Description

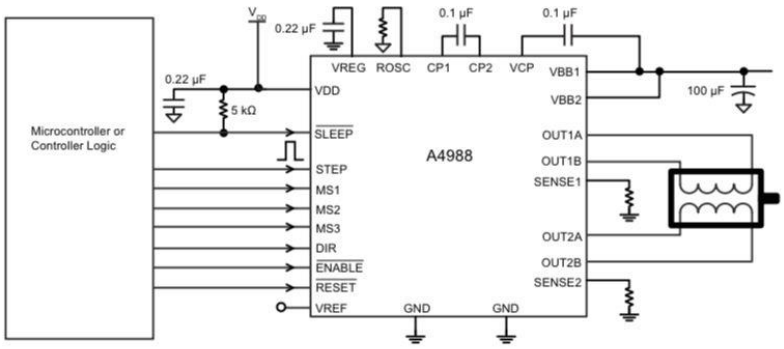
The A4988 is a complete microstepping motor driver with built-in translator for easy operation. It is designed to operate bipolar stepper motors in full-, half-, quarter-, eighth-, and sixteenth-step modes, with an output drive capacity of up to 35 V and ±2 A. The A4988 includes a fixed off-time current regulator which has the ability to operate in Slow or Mixed decay modes.

The translator is the key to the easy implementation of the A4988. Simply inputting one pulse on the STEP input drives the motor one microstep. There are no phase sequence tables, high frequency control lines, or complex interfaces to program. The A4988 interface is an ideal fit for applications where a complex microprocessor is unavailable or is overburdened.

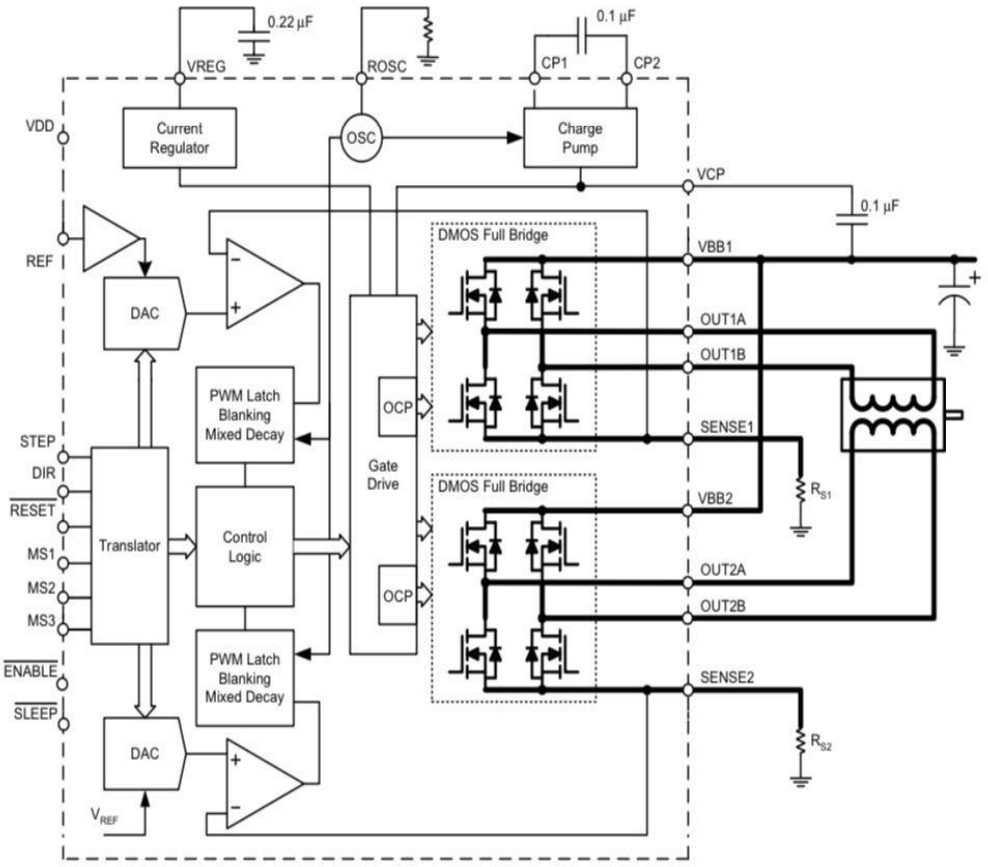
During stepping operation, the chopping control in the A4988 automatically selects the current decay mode, Slow or Mixed. In Mixed decay mode, the device is set initially to a fast decay for a proportion of the fixed off-time, then to a slow decay for the remainder of the off-time. Mixed decay current control results in reduced audible motor noise, increased step accuracy, and reduced power dissipation.

Continued on the next page…

Typical Application Diagram

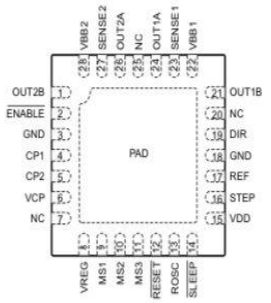


Functional Block Diagram



Allegro MicroSystems, Inc.
115 Northeast Cutoff
Worcester, Massachusetts 01615-0036 U.S.A.
1.508.853.5000; www.allegromicro.com

Pin-out Diagram



Terminal List Table

Name	Number	Description
CP1	4	Charge pump capacitor terminal
CP2	5	Charge pump capacitor terminal
VCP	6	Reservoir capacitor terminal
VREG	8	Regulator decoupling terminal
MS1	9	Logic input
MS2	10	Logic input
MS3	11	Logic input
$\overline{\text{RESET}}$	12	Logic input
ROSC	13	Timing set
$\overline{\text{SLEEP}}$	14	Logic input
VDD	15	Logic supply
STEP	16	Logic input
REF	17	G_m reference voltage input
GND	3, 18	Ground*
DIR	19	Logic input
OUT1B	21	DMOS Full Bridge 1 Output B
VBB1	22	Load supply
SENSE1	23	Sense resistor terminal for Bridge 1
OUT1A	24	DMOS Full Bridge 1 Output A
OUT2A	26	DMOS Full Bridge 2 Output A
SENSE2	27	Sense resistor terminal for Bridge 2
VBB2	28	Load supply
OUT2B	1	DMOS Full Bridge 2 Output B
$\overline{\text{ENABLE}}$	2	Logic input
NC	7, 20, 25	No connection
PAD	–	Exposed pad for enhanced thermal dissipation*

*The GND pins must be tied together externally by connecting to the PAD ground plane under the device.

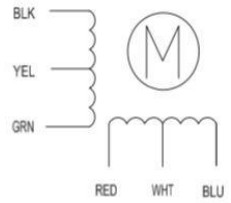
Allegro MicroSystems, Inc.
115 Northeast Cutoff
Worcester, Massachusetts 01615-0036 U.S.A.
1.508.853.5000; www.allegromicro.com

Data *Sheet Motor Stepper* NEMA 17

HIGH TORQUE HYBRID STEPPING MOTOR SPECIFICATIONS

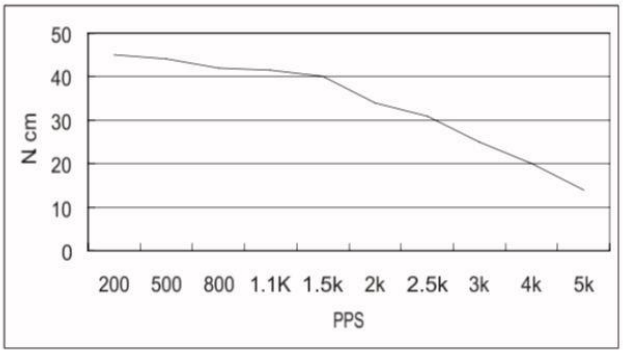
General specifications		Electrical specifications	
Step Angle (°)	1.8	Rated Voltage (V)	4
Temperature Rise (°C)	80 Max (rated current.2 phase on)	Rated Current (A)	1.2
Ambient temperature (°C)	-20～+50	Resistance Per Phase (±10%Ω)	3.3 (25°C)
Number of Phase	2	Inductance Per Phase (±20%mH)	2.8
Insulation Resistance	100MΩ, Min (500VDC)	Holding Torque (Kg.cm)	3.17
Insulation Class	Class B	Detent Torque (g.cm)	200
Max.radial force (N)	28 (20mm from the flange)	Rotor Inertia (g. cm²)	68
Max.axial force (N)	10	Weight (Kg)	0.365

● Wiring Diagram:

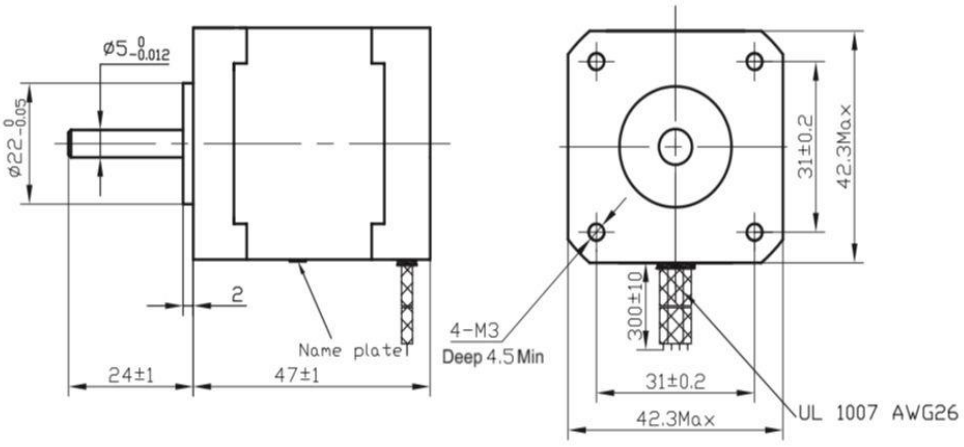


● Pull out torque curve:

VOLTAGE: 24VDC, CONSTANT CURRENT: 1.2A, HALF STEP



● Dimensions:
(unit=mm)



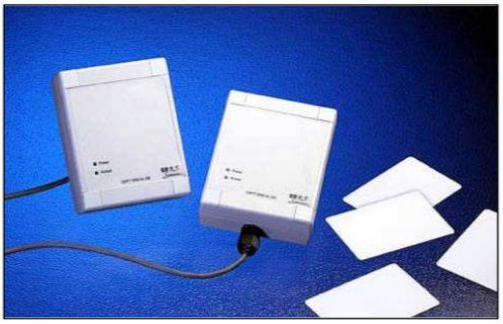
REV	REVISIONS	DESCRIPTION	BY	DATE	SY42STH47-1206A	TECHNICAL CONDITIONS
DRAW	任飞飞 2010.06.29				CHANGZHOU SONGYANG MACHINERY & ELECTRONICS NEW TECHNIC INSTITUTE	060047000
CHECK						
APPROVE						

Data *Sheet* Sensor RFID

RFID PROXIMITY ACCESS CONTROL

RFID231C-SYS

Features



- Complete Proximity Access System Kit
- Transponder Cards Read Up To 4 Inches Range With Internal Aerial.
- System Includes
  - Control & Sensor Units
  - 10 Transponder Cards.
  - Power Supply Unit
  - Connecting Cable
  - Door Strike
  - All Screws & Fixings
- Exit Switch Option
- Programmable up to 50 Cards.
- Led Indicators For Power & Accept/Reject.
- Relay Contacts 2A@12Vdc.
- User Programmable Relay, 0.5 Secs To 10 Mins
- Easy Learn, Erase Cards Via Internal Switch.
- RS232 Serial Coms to PC.
- Real Time Data Logging Output.

Description

The RFID Proximity Access System provides controlled access to premises by means of high-security transponder card technology. These are similar to credit cards in size and appearance, but no physical contact is required with the reader for the card to be scanned.

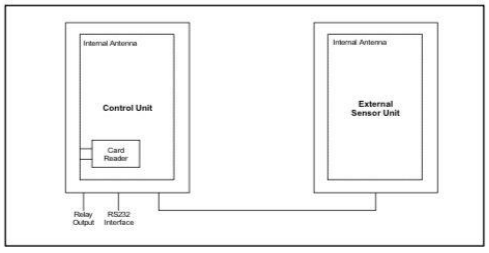
When an authorised card is placed within 4 inches of the sensor, the relay output will operate for a preset time. The system consists of a control unit and a remote sensor unit that can be fitted to the outside of a building. The control unit simply requires power and relay connections to operate. The sensor unit connects to the control unit with the cable provided.

The system has the capacity to learn up to 50 transponder cards. Programming is achieved using either a pushbutton inside the control unit, or from PC connected to the Control Unit RS232 serial port (only two connections required).

Connection to a PC enables many other functions including; control of the relay output from the PC, data logging information of which transponder card has operated the system.

The system is supplied in an ABS plastic enclosures and are IP65 rated.

Block diagram



RFID PROXIMITY ACCESS CONTROL

RFID231C-SYS

System Installation

A full installation guide is provided with the system (DS000230). This is also available from our website.

Technical specification		
Features	**Specification**	**Comments**
Protocol	Hitag2	Philips high security Protocol
User Range	Upto 15cm	Using standard Transponder card
Maximum No of Cards	50	
User feedback	LED	Data Accept / Reject LED Indication on Sensor and Control Unit
Programming Features	Yes	Via 2 onboard Switches or PC control Interface
Relay output	2A @ 12Vdc	Standard British Telecom BT47W/6 Changeover Contacts
Relay Operation Time	½ Sec – 10 mins	Programmable via onboard Switches or PC
PC Interface	RS232	2 wires only!
PC Commands	9	Complete Configuration through PC is possible.
Data Logging Output	Yes	Real time through RS232
Mounting	Wall or Pole	With Screw fixings provided
Environmental Protection	IP65	
Enclosure Material	ABS	
Power Supply: UK Version	UK 230Vac	UK Plug top
Power Supply: INTL Version	INTL 110 –240Vac	Standard International IEC Input
Power Consumption	~50mA	Quiescent
	~90mA	Operating Relay
Dimensions	110 x 85 x 35mm	
Operating Temp	-0 to + 70 °C	
Storage Temp	-55 to + 125 °C	

Technical specification: Transponder Card		
Features	**Specification**	**Comments**
Card Type	ISO Hitag2	Can be any Hitag2 card package
Material	White PVC	May be Custom Silk Screened
Size		As standard credit card

PART No	DESCRIPTION
RFID231C-SYS	RFID Proximity Access System with 10 Cards
RFCARDHITAG2	Spare RFID Card tag (Pack of 10)

Should you require further assistance, please call;

R. F. Solutions Ltd,
Unit 21, Cliffe Ind Est, South St, Lewes, E Sussex, BN8 6JL. England.
Tel +44 (0)1273 898 000. Fax +44 (0)1273 480 661.
Email sales@rfsolutions.co.uk
http://www.rfsolutions.co.uk

RF Solutions is a member of the Low Power Radio Association.

ISSN 2477-2755 (P) / 2622-2981 (E)

SURAT KETERANGAN PENERIMAAN NASKAH
No. 33/Ampere/VIII/2023

Dewan Redaksi Jurnal Ampere, menyatakan bahwa:

Judul	**Implementasi DOF (Degree Of Freedom) Pada Pergerakan Motor Stepper Smart Inventory 3 Axis**
Penulis	**Ali Ripaldo1, Yordan Hasan2*, Johansyah Al Rasyid3**
Afiliasi	**Prodi Teknik Elektro Mekatronika, Jurusan Teknik Elektro, Politeknik Negeri Sriwijaya, Indonesia**

Adalah benar telah mengirimkan artikelnya ke Jurnal Ampere dan saat ini sedang dalam proses **PEER REVIEW** sesuai dengan prosedur Jurnal Ampere Universitas PGRI Palembang,

Demikian Surat Keterangan ini dibuat, untuk dapat dipergunakan sebagaimana perlunya.

Palembang, 7 Agustus 2022
Redaksi Jurnal Ampere

Nita Nurdiana, MT

Jurnal Ampere
Prodi Teknik Elektro Fakultas Teknik Universitas PGRI Palembang
Jl. A. Yani Lrg. Gotong Royong 9-10 Ulu Palembang
Email : ampereupgrip@gmail.com