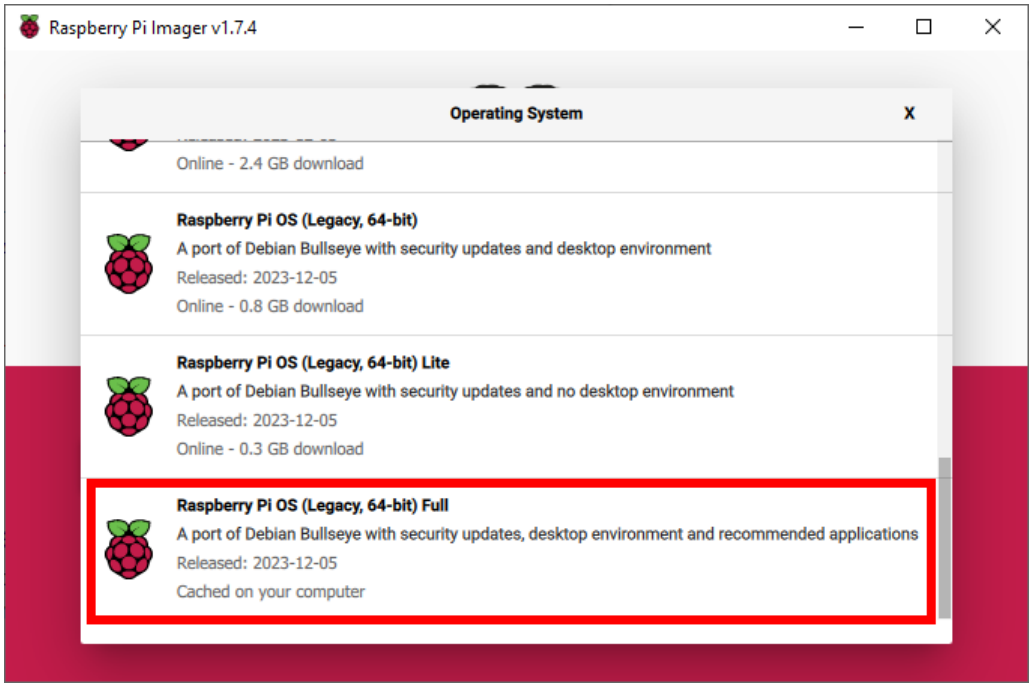
ขั้นตอนการสร้างหุ่นยนต์ Teacher Robot

ติดตั้งระบบปฏิบัติการ Raspberry Pi OS (Legacy, 64-bit) 64 Full

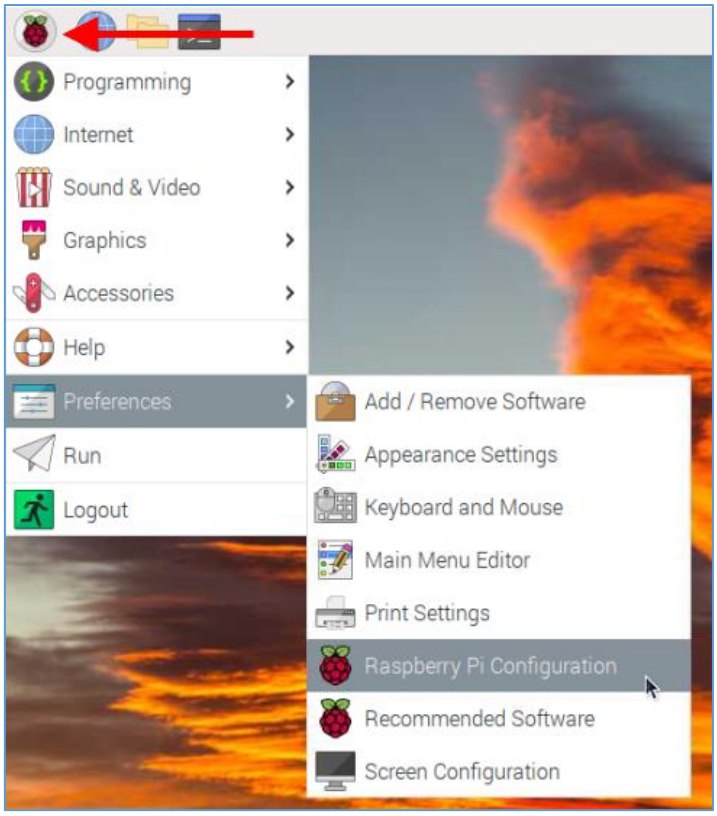
1. เสียบ SD Card เข้ากับเครื่องคอมพิวเตอร์ จากนั้นเปิดโปรแกรม Raspberry Pi Imager ติดตั้งระบบปฏิบัติการ Raspberry Pi OS (Legacy,64-bit) Full

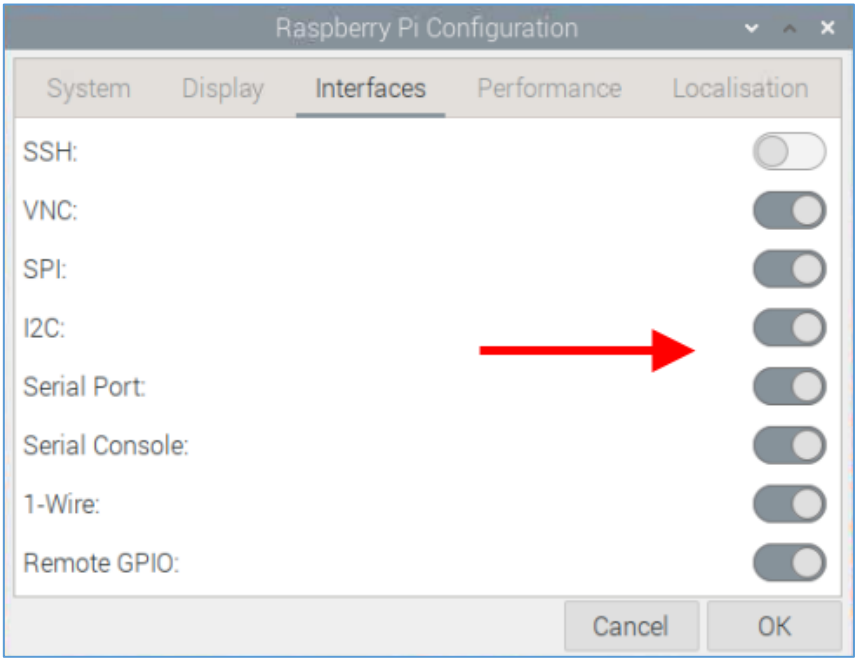


2. เมื่อติดตั้งเสร็จแล้ว ให้ถอด SD Card จากเครื่องคอมพิวเตอร์เสียบเข้าบอร์ด Raspberry Pi พร้อมทำตามขั้นตอนที่ระบบแนะนำ ระบบจะบู๊ตเข้าหน้า Desktop

ตั้งค่าบอร์ด Raspberry Pi

1. เปิดใช้งานอินเตอร์เฟซ โดยคลิกที่ Pi ---> Preference ---> Raspberry Pi Configuration จากนั้นคลิกแท็บ Interfaces จากนั้นคลิกเปิดใช้งานอินเตอร์เฟซ ดังรูป





2. เปิดใช้งาน VNC โดยเปิด Terminal จากนั้นพิมพ์คำสั่ง sudo raspi-config ---> Interface Option ---> VNC ---> Enable

ติดตั้งซอฟต์แวร์ต่าง ๆ

1. ติดตั้งซอฟต์แวร์ที่ระบบแนะนำ โดยคลิกที่ Pi ---> Preference ---> Recommended Software จากนั้นคลิกเลือกโปรแกรมที่ต้องการติดตั้ง แล้วคลิกปุ่ม Apply เพื่อติดตั้ง

2. ติดตั้งโปรแกรม Arduino IDE โดยเปิด Terminal แล้วพิมพ์คำสั่ง

```
sudo apt-get install arduino
```

3. ติดตั้ง PyQt5 สำหรับออกแบบหน้าจอ GUI โดยเปิด Terminal แล้วพิมพ์คำสั่ง

```
sudo apt-get install qttools5-dev-tools
```

การเชื่อมต่อหุ่นยนต์กับบอร์ด Arduino Mega 2560

1. หลอดไฟ RGB (ถ้าทั้ง 3 สีมีค่าเป็น 1 จะได้สีขาว)

- เส้นสีดำ → GND (Arduino)
- เส้นสีแดง (R) → D11
- เส้นสีเขียว (G) → D12
- เส้นสีน้ำเงิน (B) → D13

2. ลำโพงใหญ่ด้านหน้า

- เส้นสีแดง → D45
- เส้นสีดำ → GND (Arduino)

3. DC Motor แบบ 6 เส้น (แขนซ้าย)

- เส้นที่ 1 สีแดง VCC → 12 โวลต์จากแบตเตอรี่
- เส้นที่ 2 สีดำ GND → GND
- เส้นที่ 3 สีเขียว → D30
- เส้นที่ 4 สีเหลือง → D31
- เส้นที่ 5 สีส้ม → D32
- เส้นที่ 6 สีน้ำเงิน → D5

4. DC Motor แบบ 6 เส้น (แขนขวา)

- เส้นที่ 1 สีแดง VCC ⟶ 12 โวลต์จากแบตเตอรี่
- เส้นที่ 2 สีดำ GND ⟶ GND
- เส้นที่ 3 สีเขียว ⟶ D34
- เส้นที่ 4 สีเหลือง ⟶ D35
- เส้นที่ 5 สีส้ม ⟶ D39
- เส้นที่ 6 สีน้ำเงิน ⟶ D6

5. DC Motor แบบ 6 เส้น (ล้อซ้าย)

- เส้นที่ 1 สีแดง VCC ⟶ 12 โวลต์จากแบตเตอรี่
- เส้นที่ 2 สีดำ GND ⟶ GND
- เส้นที่ 3 สีเหลือง ⟶ D36
- เส้นที่ 4 สีเขียว ⟶ D37
- เส้นที่ 5 สีส้ม ⟶ D38
- เส้นที่ 6 สีน้ำเงิน ⟶ D9

6. DC Motor แบบ 6 เส้น (ล้อขวา)

- เส้นที่ 1 สีแดง VCC ⟶ 12 โวลต์จากแบตเตอรี่
- เส้นที่ 2 สีดำ GND ⟶ GND
- เส้นที่ 3 สีเหลือง ⟶ D23
- เส้นที่ 4 สีเขียว ⟶ D25
- เส้นที่ 5 สีส้ม ⟶ D26
- เส้นที่ 6 สีน้ำเงิน ⟶ D8

7. DC Motor แบบ 6 เส้น (ล้อหลัง)

- เส้นที่ 1 สีแดง VCC ⟶ 12 โวลต์จากแบตเตอรี่
- เส้นที่ 2 สีดำ GND ⟶ GND

- เส้นที่ 3 สีเหลือง ⟶ D29

- เส้นที่ 4 สีเขียว ⟶ D28

- เส้นที่ 5 สีส้ม ⟶ D27

- เส้นที่ 6 สีน้ำเงิน ⟶ D10

8. IR Sensor 6 ตัวที่ฐาน

- เส้นสีดำ ⟶ GND

- เส้นสีแดง ⟶ VCC 5 V

- เส้นสีเขียว ⟶ SDA

- เส้นสีเหลือง ⟶ SCL

9. MUX

- เส้นที่ 1 ⟶ GND

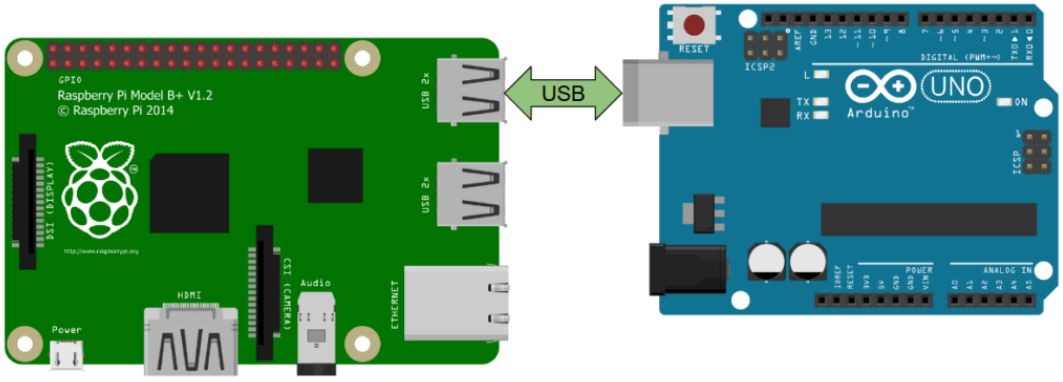
- เส้นที่ 2 ⟶ VCC 5 V

- เส้นที่ 3 D20 ⟶ SDA

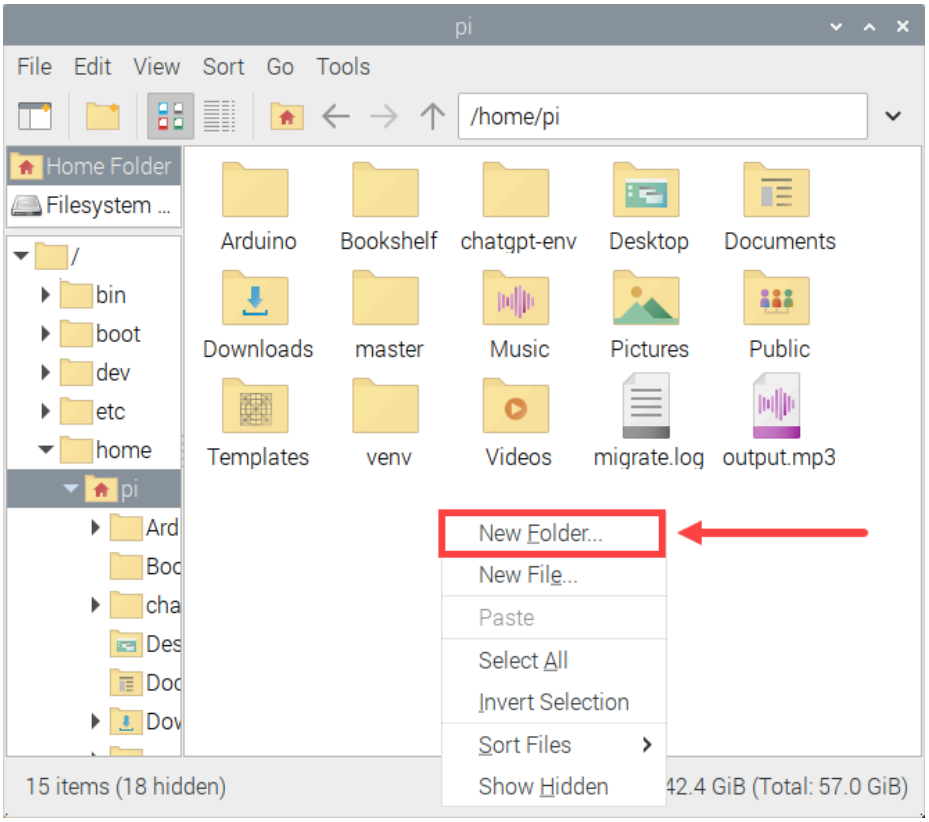
- เส้นที่ 4 D21 ⟶ SCL

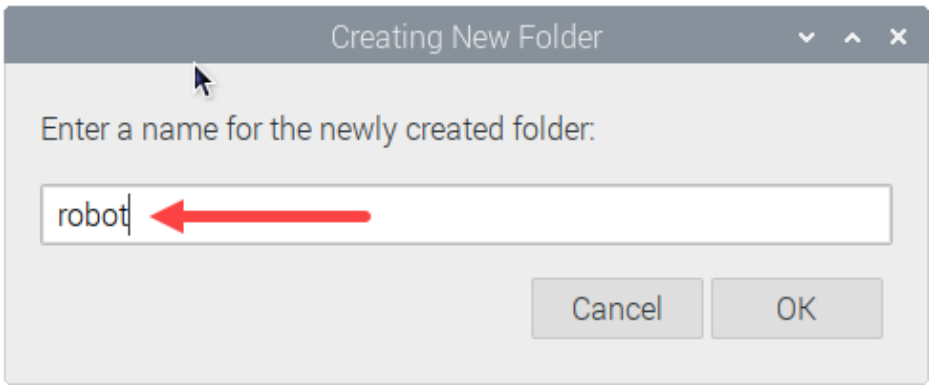
การเขียนคำสั่งด้วยบอร์ด Arduino Mega 2560

1. เชื่อมต่อบอร์ด Arduino Mega 2560 กับพอร์ต USB ของบอร์ด Raspberry Pi



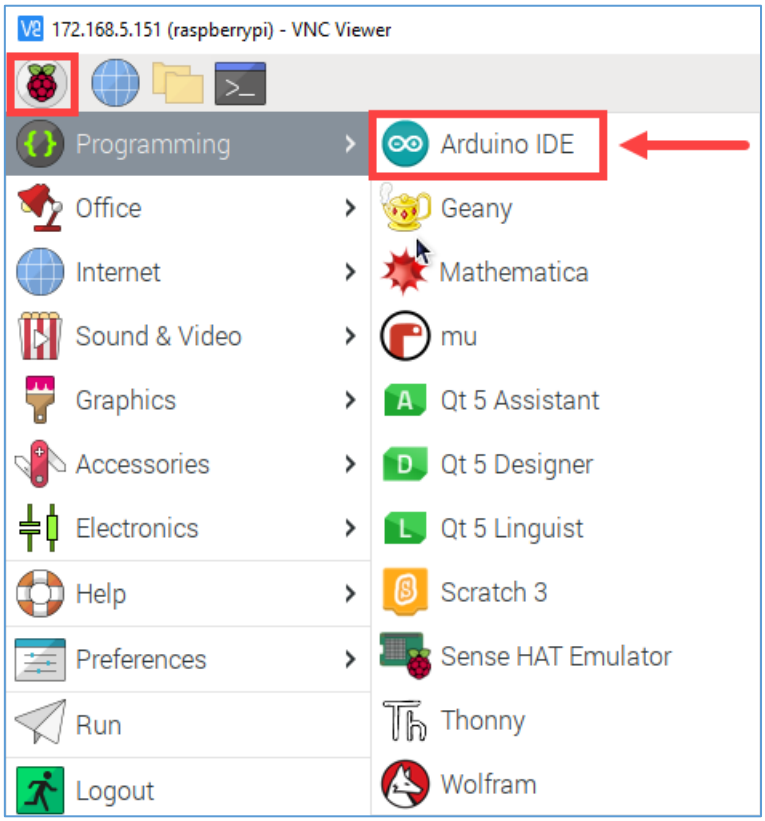
2. สร้างโฟลเดอร์สำหรับเก็บไฟล์ต่างๆ ของโปรแกรม โดยตั้งชื่อโฟลเดอร์เป็น robot



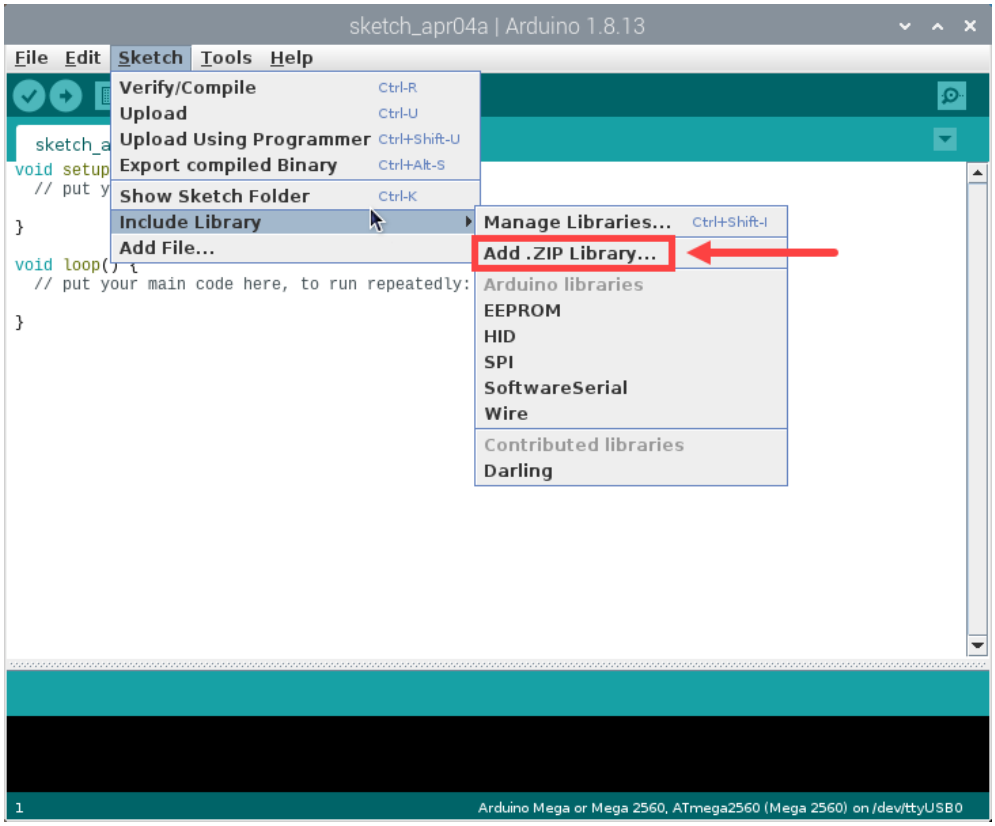


2. นำเข้าไลบรารี่

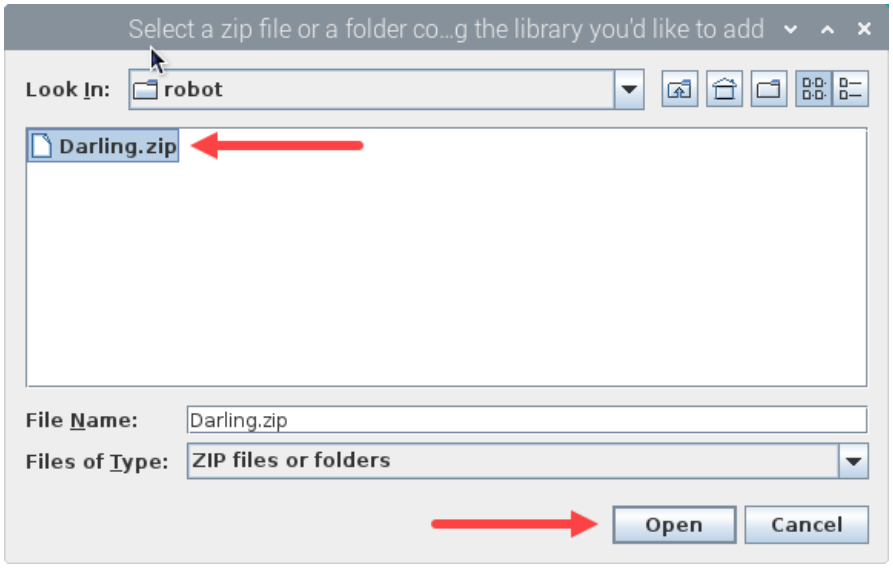
2.1 เปิดโปรแกรม Arduino IDE



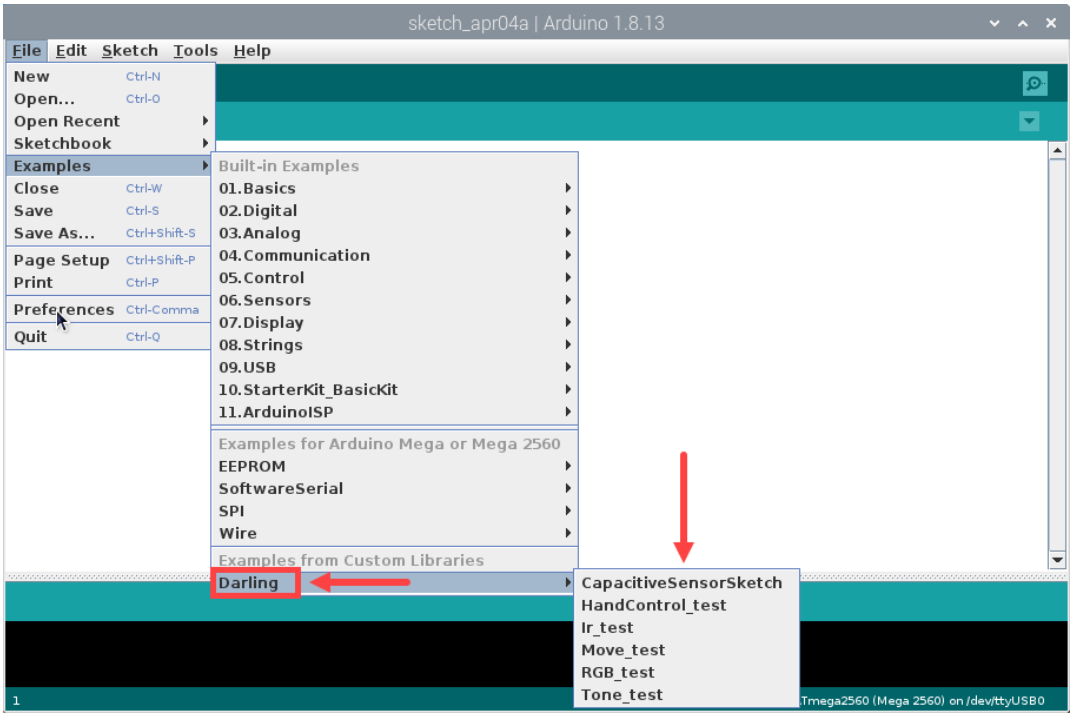
2.2 คลิกที่ Sketch ---> Include Library ---> Add .ZIP Library…



2.3 คลิกเลือกไฟล์ Darling.zip



2.4 คลิกที่ File ---> Examples จะปรากฏไลบรารี่ Darling ให้พร้อมใช้งาน



3. เขียนโปรแกรมด้วยภาษา C โดยระบุฟังก์ชันการทำงานของหุ่นยนต์ ดังนี้

```
#include <Wire.h>
#include <Arduino.h>
#include "MotorMove.h"
#include "Hand_robot.h"
#include "RGB_robot.h"

Hand_robot handLeft = Hand_robot(30, 31, 32, 5, LEFT);
Hand_robot handRight = Hand_robot(34, 35, 39, 6, RIGHT);
RGB_robot rgb = RGB_robot(11,12,13);

MotorMove motor = MotorMove();

void setup(){
  Serial.begin(9600);
  motor.Begin();
  handLeft.begin();
  handRight.begin();
  rgb.begin();
}

void Hand_up(){
  handLeft.HandRotateWithSpeed("fwd", 100, 1000);
  handRight.HandRotateWithSpeed("fwd", 100, 1000);
  delay(1000);
}

void Hand_down(){
  handLeft.HandRotateWithSpeed("back", 100, 1000);
  handRight.HandRotateWithSpeed("back", 100, 1000);
  delay(1000);
}

void Hand_walk(){
  for (int i = 0; i < 6; i++){
    handLeft.HandRotate("fwd");
    handRight.HandRotate("back");
    handLeft.HandRotate("back");
    handRight.HandRotate("fwd");
  }
}

void Hand_stop(){
  handLeft.Hand_Stop();
  handRight.Hand_Stop();
}

void Move_forward(){
  motor.MoveForwardWithDelay(80, 1000);
  motor.MoveStop();
  delay(1000);
}

```

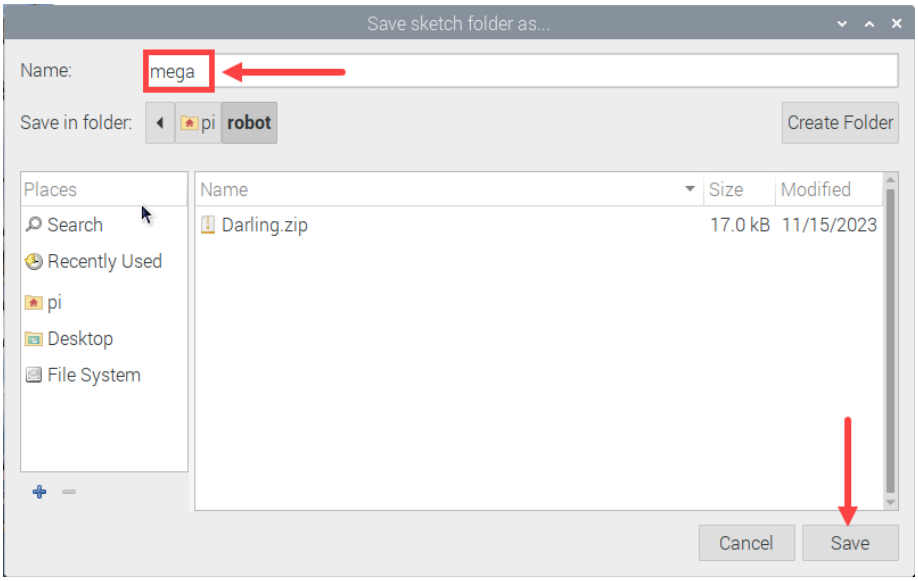
```
53 void Move_backward(){
54   motor.MoveBackwardWithDelay(80, 500);
55   motor.MoveStop();
56   delay(1000);
57 }
58
59 void Turn_left(){
60   motor.TurnLeftWithDelay(80, 500);
61   motor.MoveStop();
62   delay(1000);
63 }
64
65 void Turn_right(){
66   motor.TurnRightWithDelay(80, 500);
67   motor.MoveStop();
68   delay(1000);
69 }
70
71 void Slide_left(){
72   motor.SlideLeftWithDelay(80, 500);
73   motor.MoveStop();
74   delay(1000);
75 }
76
77 void Slide_right(){
78   motor.SlideRightWithDelay(80, 500);
79   motor.MoveStop();
80   delay(1000);
81 }
82
83 void Move_stop(){
84   motor.MoveStop();
85 }
86
87 void GreenRGB_on(){
88   rgb.fullBody(0,255,0);
89   delay(1000);
90 }
91
92 void RedRGB_on(){
93   rgb.fullBody(255,0,0);
94   delay(1000);
95 }
96
97 void BlueRGB_on(){
98   rgb.fullBody(0,0,255);
99   delay(1000);
100 }
101
102 void RGB_off(){
103   rgb.fullBody(0,0,0);
104   delay(1000);
105 }
106
```

```
109
110 void loop() {
111   if (Serial.available() > 0) {
112
113     String data = Serial.readStringUntil('\n');
114     if(data == "FW"){
115       //เดินหน้า
116       Move_forward();
117       Serial.println(data);}
118
119     else if(data == "BW"){
120       //ถอยหลัง
121       Move_backward();
122       Serial.println(data);}
123
124     else if(data == "TR"){
125       //เลี้ยวขวา
126       Turn_right();
127       Serial.println(data);}
128
129     else if(data == "TL"){
130       //เลี้ยวซ้าย
131       Turn_left();
132       Serial.println(data);}
133
134     else if(data == "SR"){
135       //สไลด์ไปทางขวา
136       Slide_right();
137       Serial.println(data);}
138
139     else if(data == "SL"){
140       //สไลด์ไปทางซ้าย
141       Slide_left();
142       Serial.println(data);}
143
144     else if(data == "STOP"){
145       //หยุดเดิน
146       Move_stop();
147       Serial.println(data);}
148
149     else if(data == "HU"){
150       //ยกแขน
151       Hand_up();
152       Serial.println(data);}
153
154     else if(data == "HD"){
155       //เอาแขนลง
156       Hand_down();
157       Serial.println(data);}
158
```

```
159     else if(data == "HW"){
160       //เหวี่ยงแขนตอนเดิน
161       Hand_walk();
162       Serial.println(data);}
163
164     else if(data == "HSTOP"){
165       //แขนหยุด
166       Hand_stop();
167       Serial.println(data);}
168
169     else if(data == "Green"){
170       GreenRGB_on();
171       Serial.println(data);}
172
173     else if(data == "Red"){
174       RedRGB_on();
175       Serial.println(data);}
176
177      else if(data == "Blue"){
178       BlueRGB_on();
179       Serial.println(data);}
180
181     else if(data == "RGBoff"){
182       RGB_off();
183       Serial.println(data);}
184   }
185 }
```

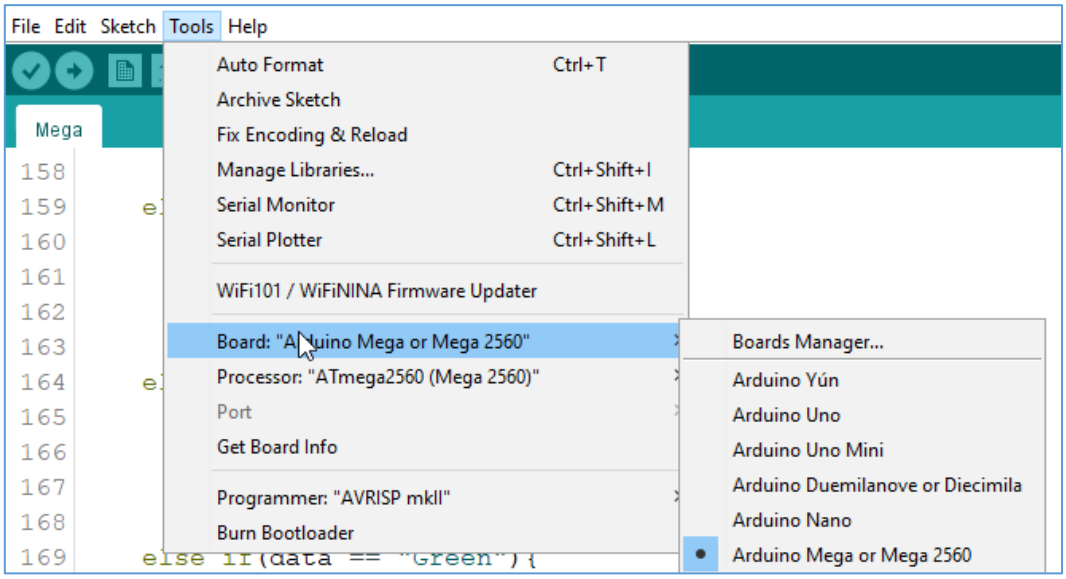
4. อัพโหลดโค้ดคำสั่งใส่บอร์ด Arduino Mega 2560

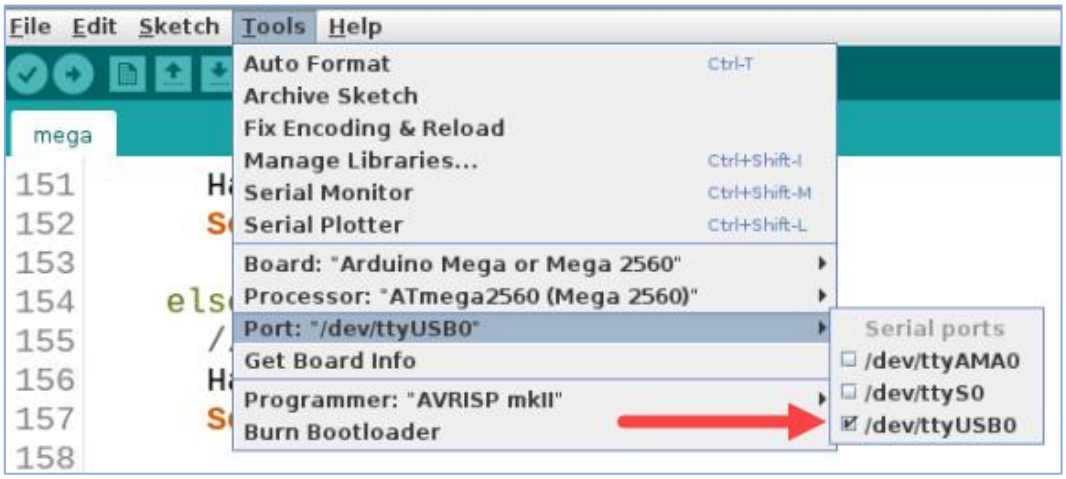
4.1 บันทึกไฟล์ในโฟลเดอร์ robot โดยตั้งชื่อไฟล์ว่า mega.ino



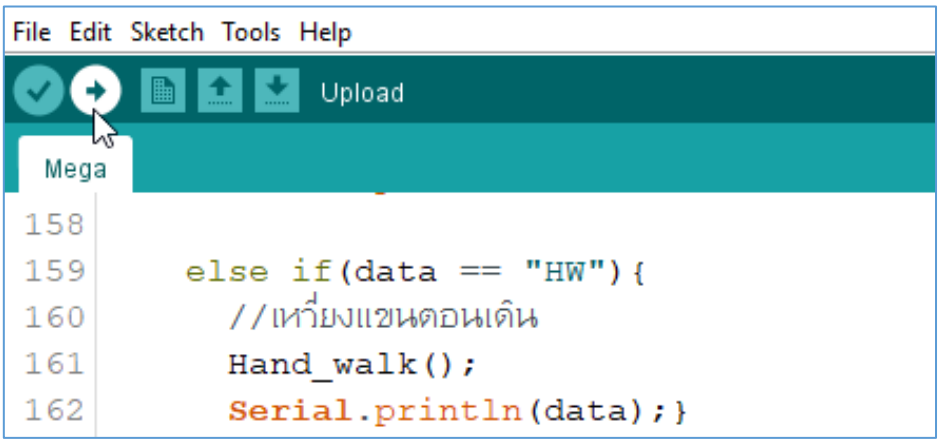
4.2 ระบุบอร์ดและ Port โดยคลิกที่ Tools ---> Board ---> Arduino Mega or Mega 2560

Port ---> /dev/ttyUSB0



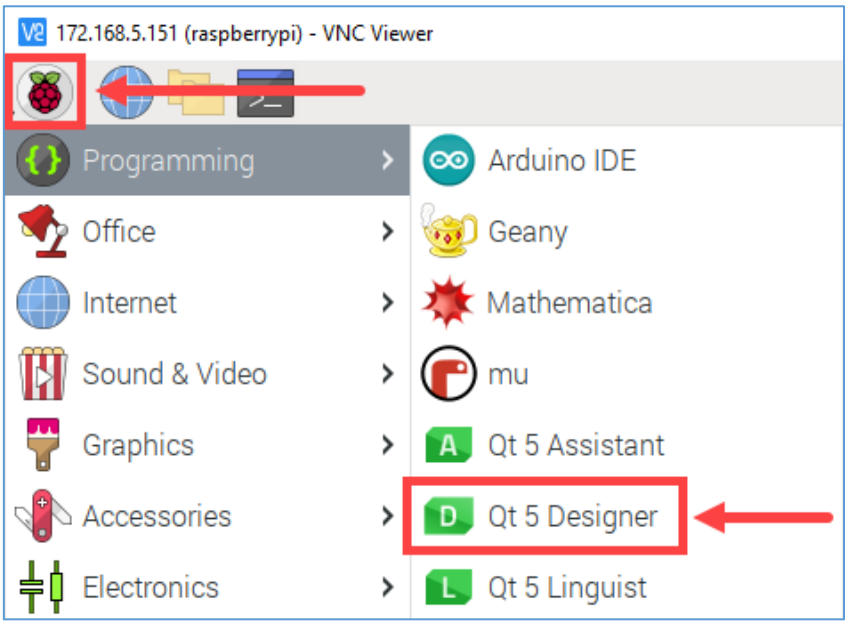


4.3 คลิก Upload



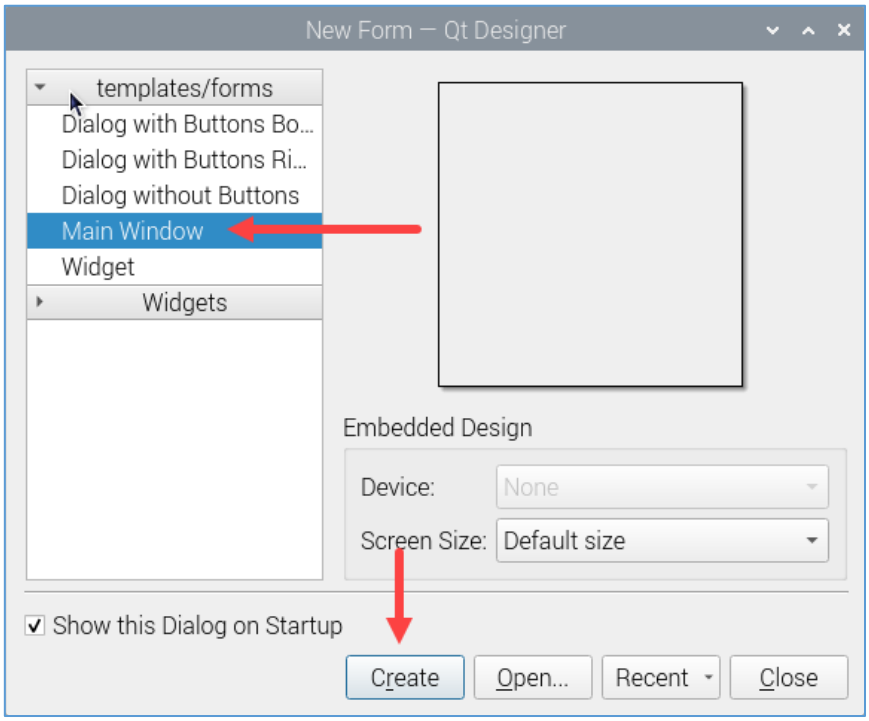
ออกแบบหน้าจอด้วยโปรแกรม PyQt Designer

1. เปิดโปรแกรม PyQt Designer เพื่อออกแบบหน้าจอ GUI โดยคลิกที่ Pi --> Programming --> Qt 5 Designer

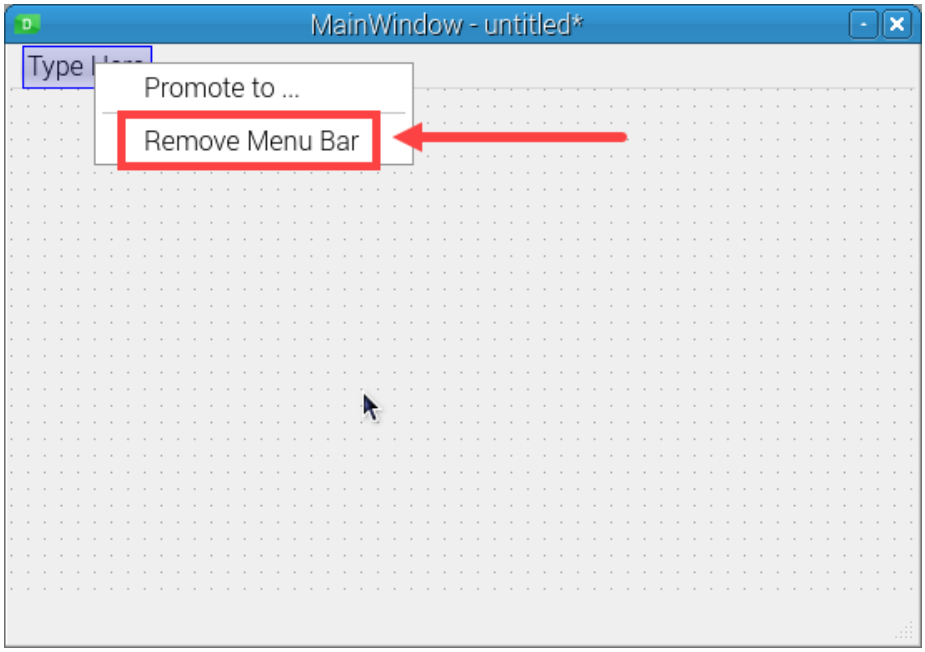


2. ออกแบบหน้าจอหลัก (main.ui)

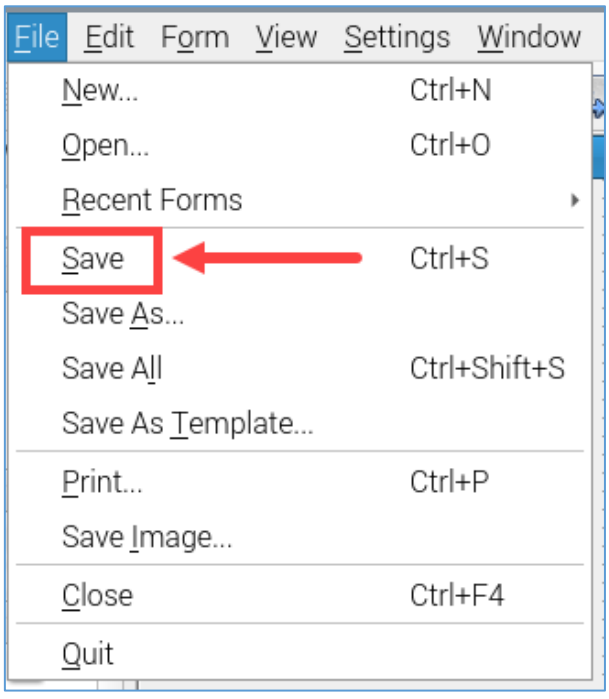
2.1 เลือกเท็มเพลต โดยคลิกเลือก Main Window แล้วคลิกปุ่ม Create

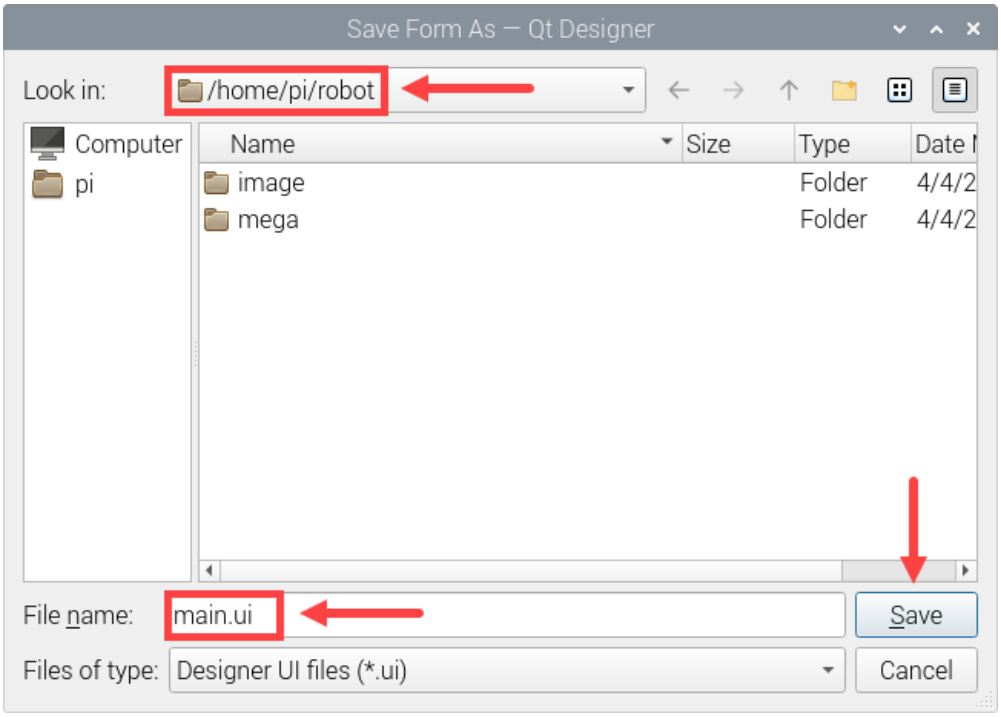


2.2 ลบแท็บ Menu Bar โดยคลิกขวาที่แท็บ Type Here จากนั้นเลือกคำสั่ง Remove Menu Bar

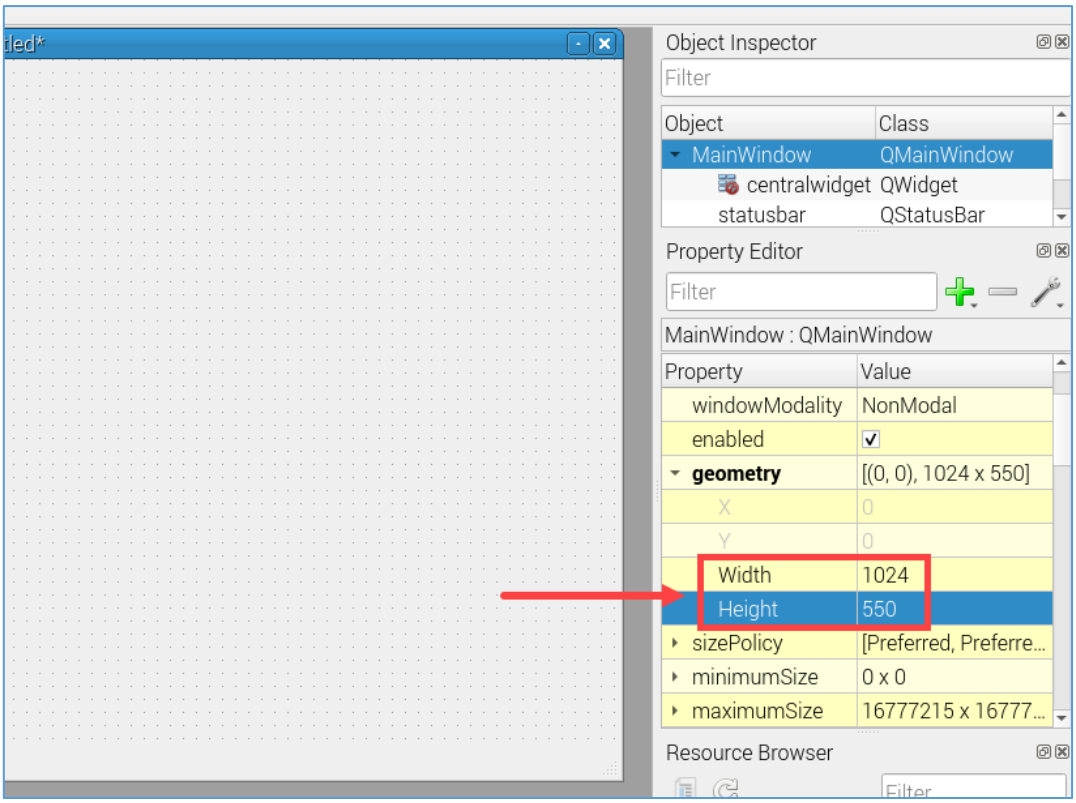


2.3 บันทึกไฟล์ ไว้ในโฟลเดอร์ robot โดยตั้งชื่อว่า “main.ui” โดยคลิกแท็บ File --> Save



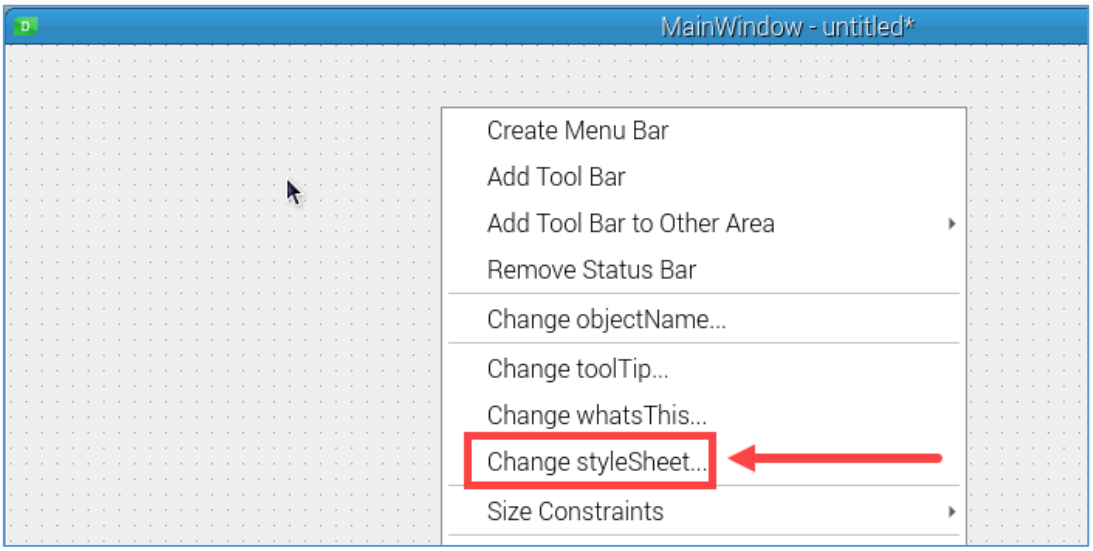


2.4 กำหนดขนาดของหน้าจอ ให้เท่ากับหน้าจอหุ่นยนต์ โดยคลิกที่ geometry แล้วกำหนดค่า Width: 1024 และ Height: 550

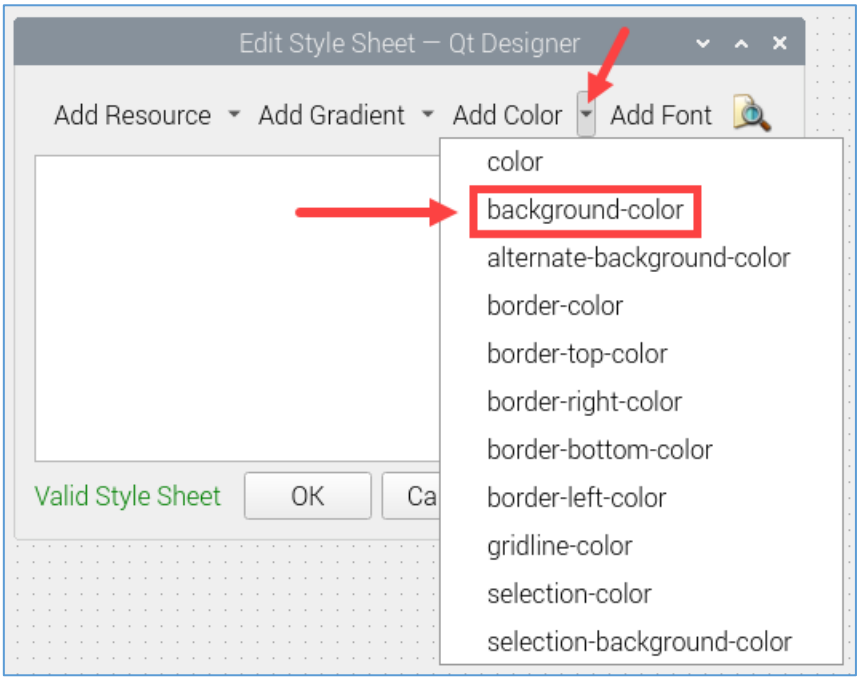


2.5 ใส่สีพื้นหลัง

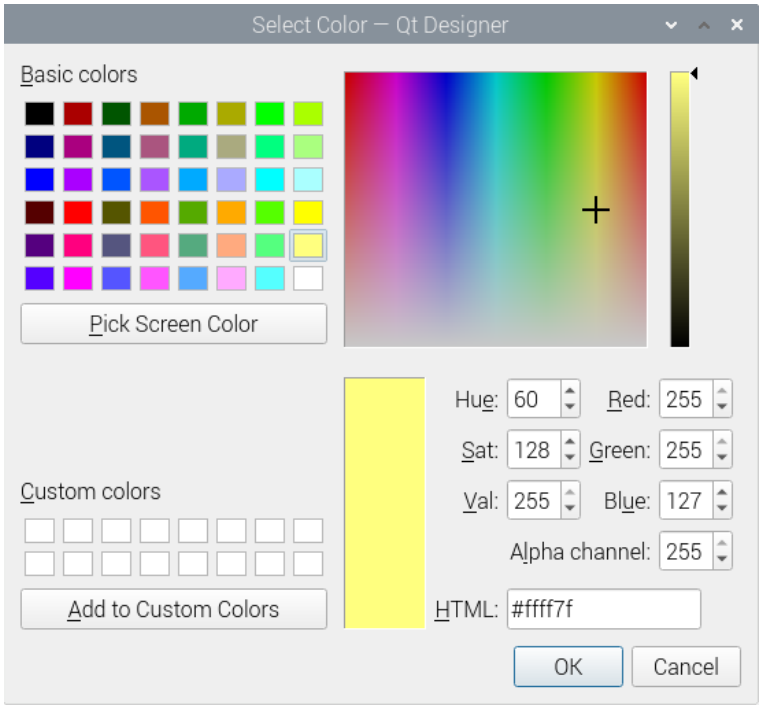
1) คลิกขวาบน MainWindow แล้วคลิกเลือก Change stylesheet



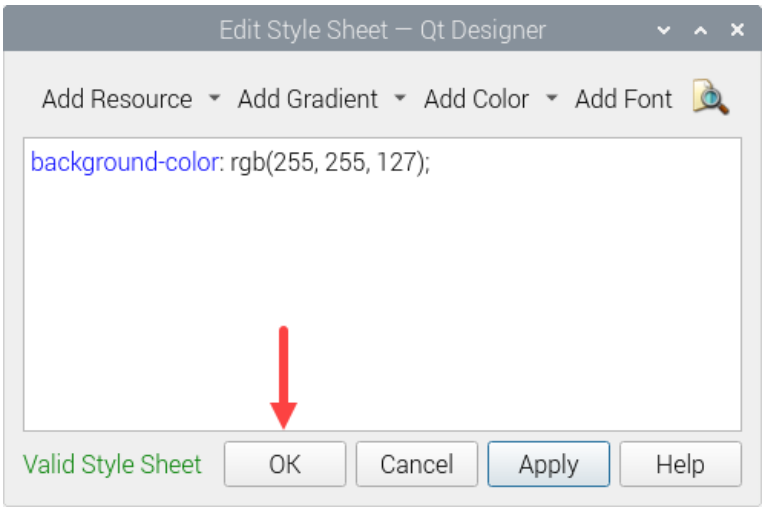
2) จากนั้นคลิกลูกศรด้านหลัง Add Color แล้วคลิกเลือก background-color



3) เลือกสีที่ต้องการ จากนั้นคลิกปุ่ม OK

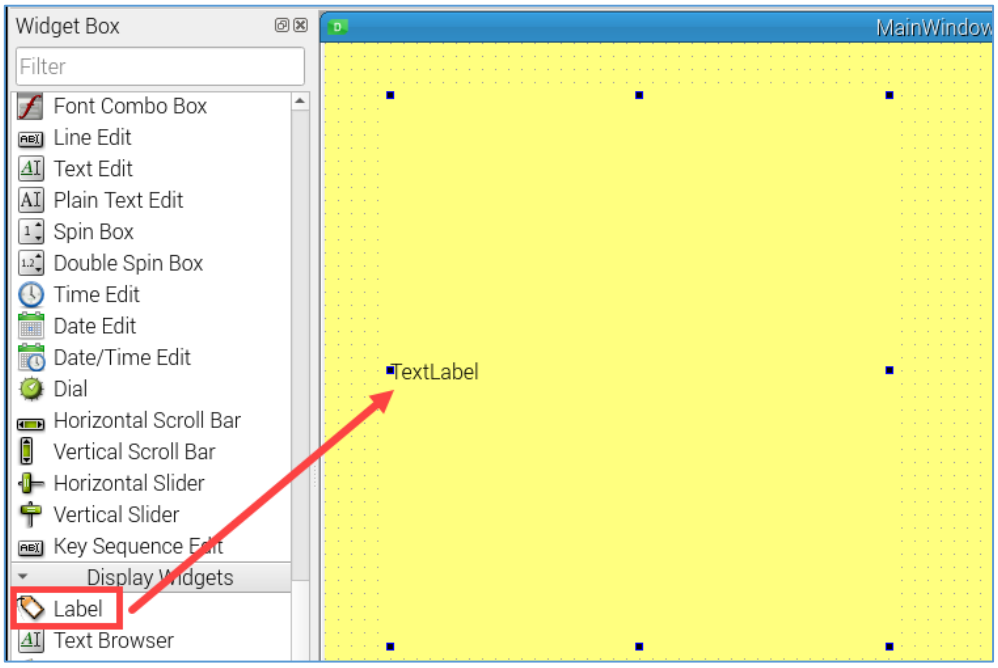


4) คลิกปุ่ม OK

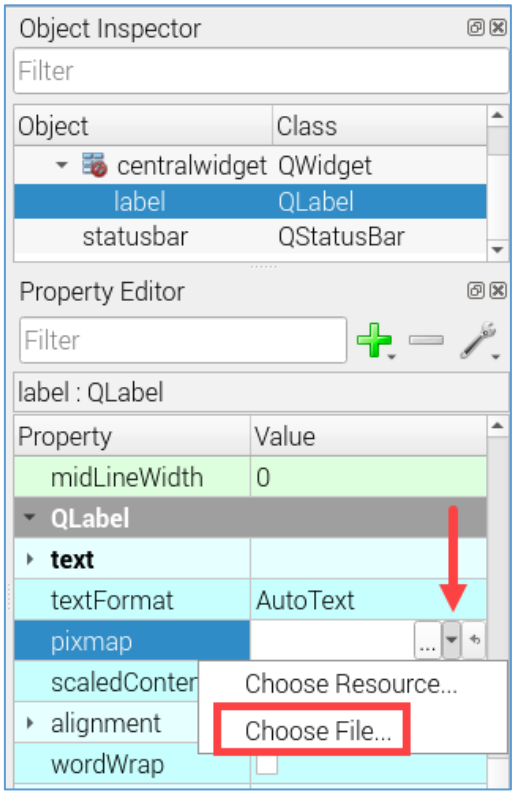


2.6 แทรกภาพหุ่นยนต์เคลื่อนไหว (.gif)

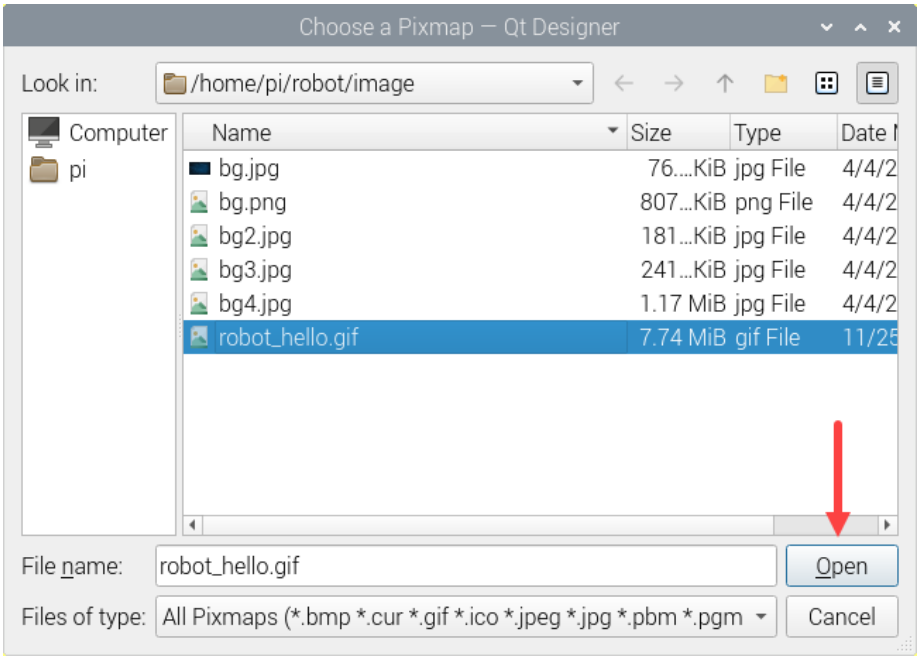
1) ลากเครื่อง Label มาวางบน MainWindow



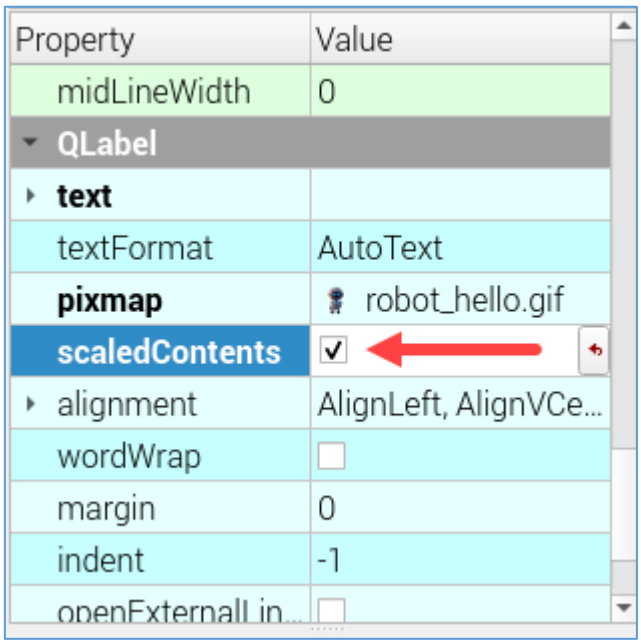
2) แก้ไข Property ของ Label โดยคลิกลูกศรด้านหลัง pixmap เลือก Choose File



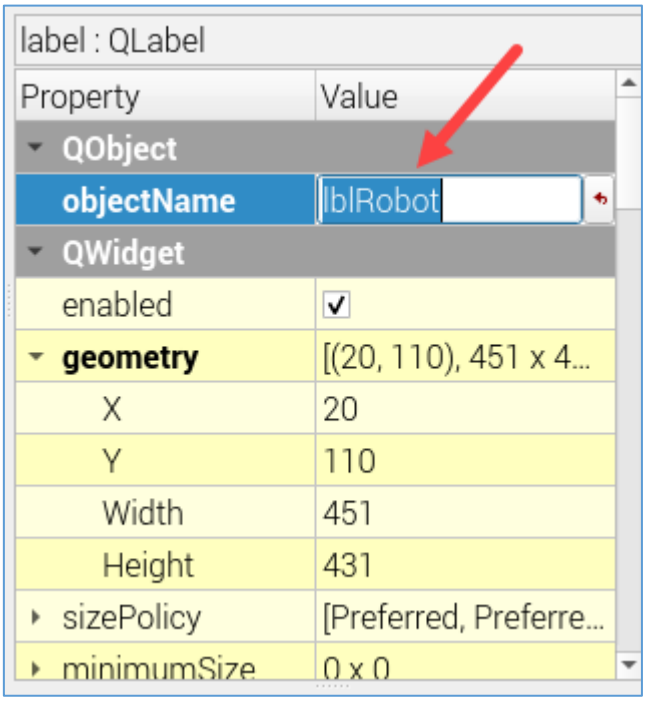
3) คลิกเลือกไฟล์รูปภาพ .gif จากนั้นคลิกปุ่ม Open



4) **ปรับรูปให้พอดีกับ Label** โดยคลิกถูกตรง scaledCentents

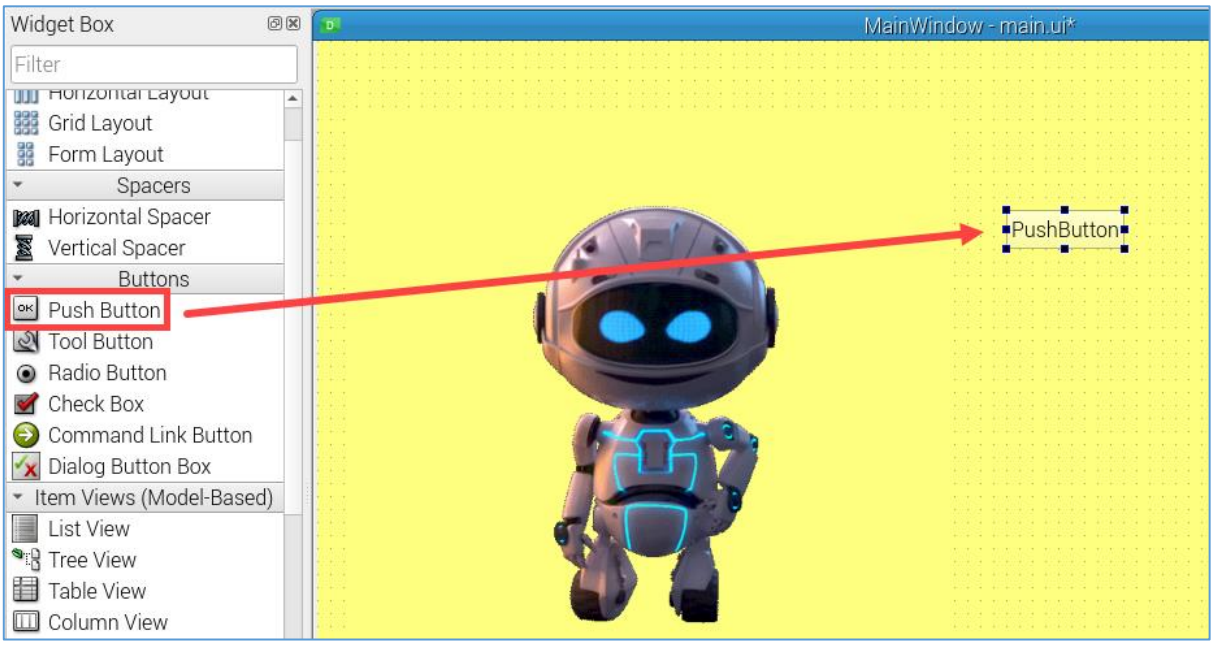


5) ตั้งชื่อ Label ว่า “lblRobot” ในช่อง objectName

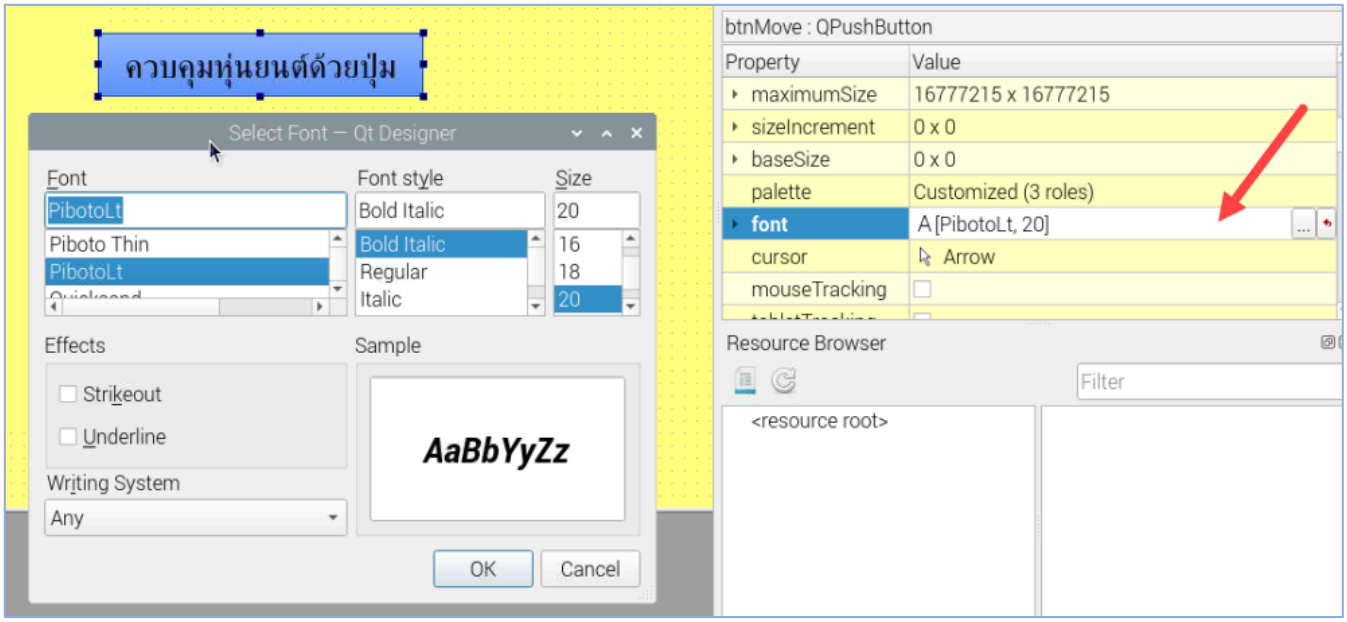


2.7 สร้างปุ่มเมนู “ควบคุมหุ่นยนต์ด้วยปุ่ม”

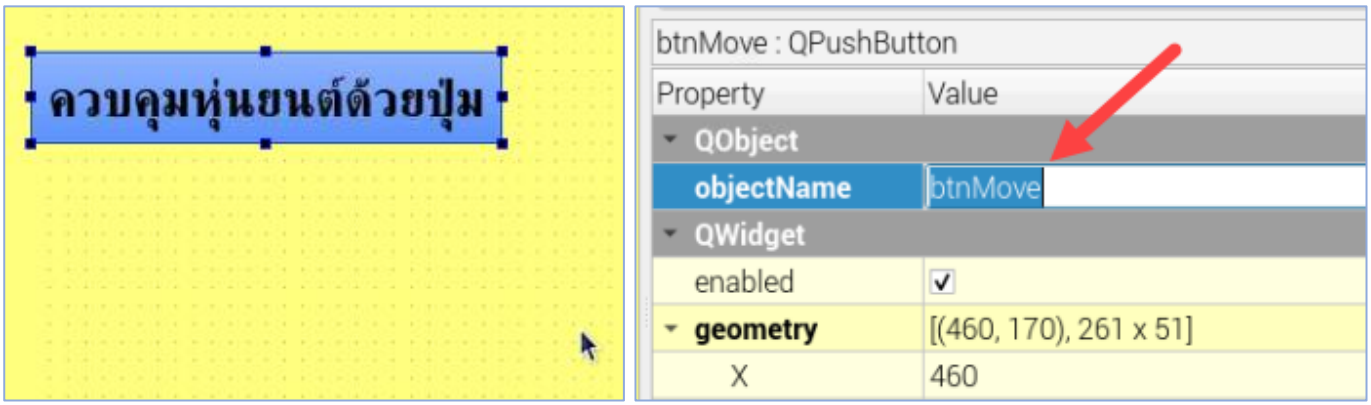
1) ลาก Push Button มาวางบน MainWindow



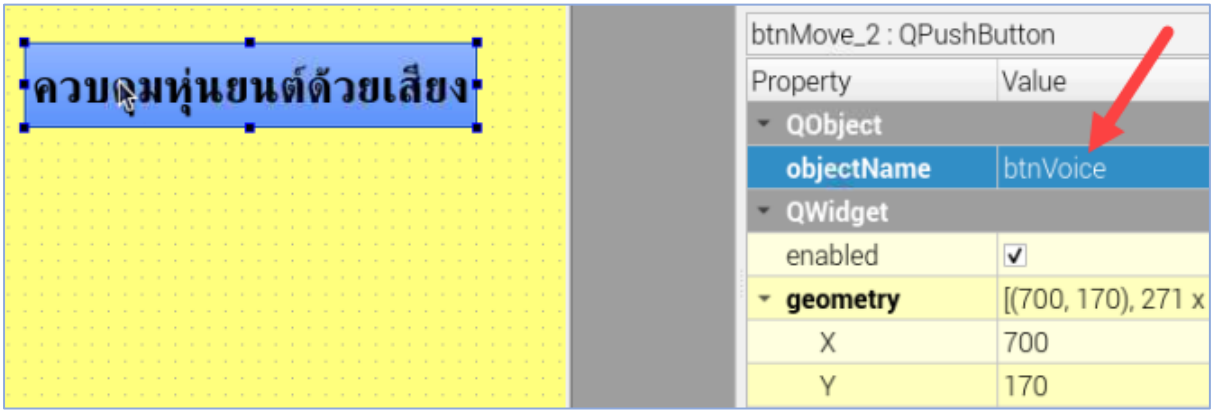
2) แก้ไขข้อความบนปุ่ม โดยดับเบิ้ลคลิกที่ปุ่ม จากนั้นพิมพ์ข้อความว่า “ควบคุมหุ่นยนต์ด้วยปุ่ม” จากนั้นแก้ไขขนาดและลักษณะของตัวอักษรใน Property: font แล้วคลิกปุ่ม OK



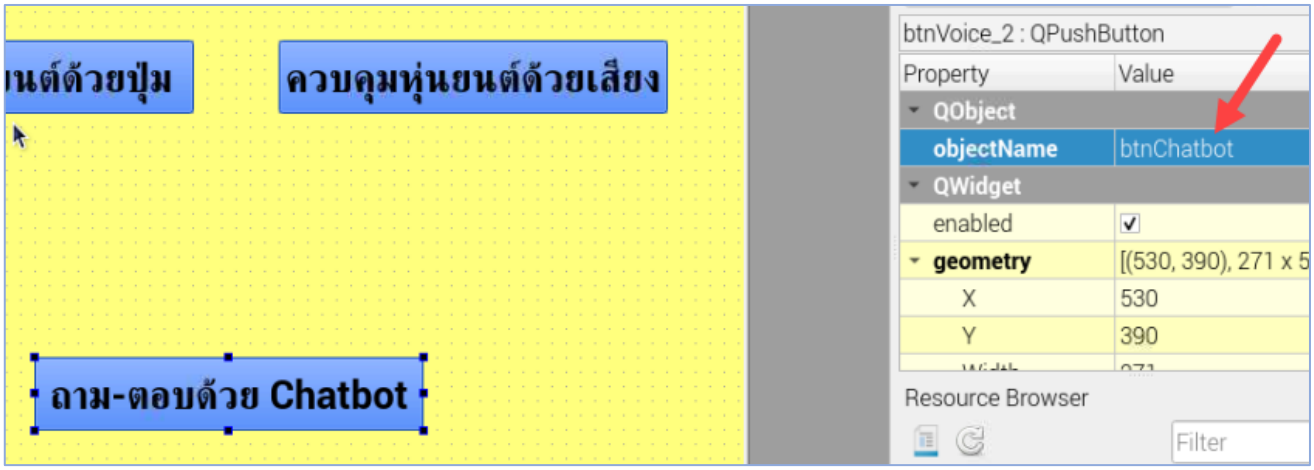
3) ตั้งชื่อปุ่ม โดยคลิกที่ objectName ตั้งชื่อเป็น “btnMove”



2.8 สร้างปุ่มเมนู “ควบคุมหุ่นยนต์ด้วยเสียง” โดยคัดลอกปุ่มแรก จากนั้นแก้ไขข้อความบนปุ่มเป็น “ควบคุมหุ่นยนต์ด้วยเสียง” แล้วเปลี่ยนชื่อปุ่มโดยคลิกที่ objectName ตั้งชื่อเป็น “btnVoice”



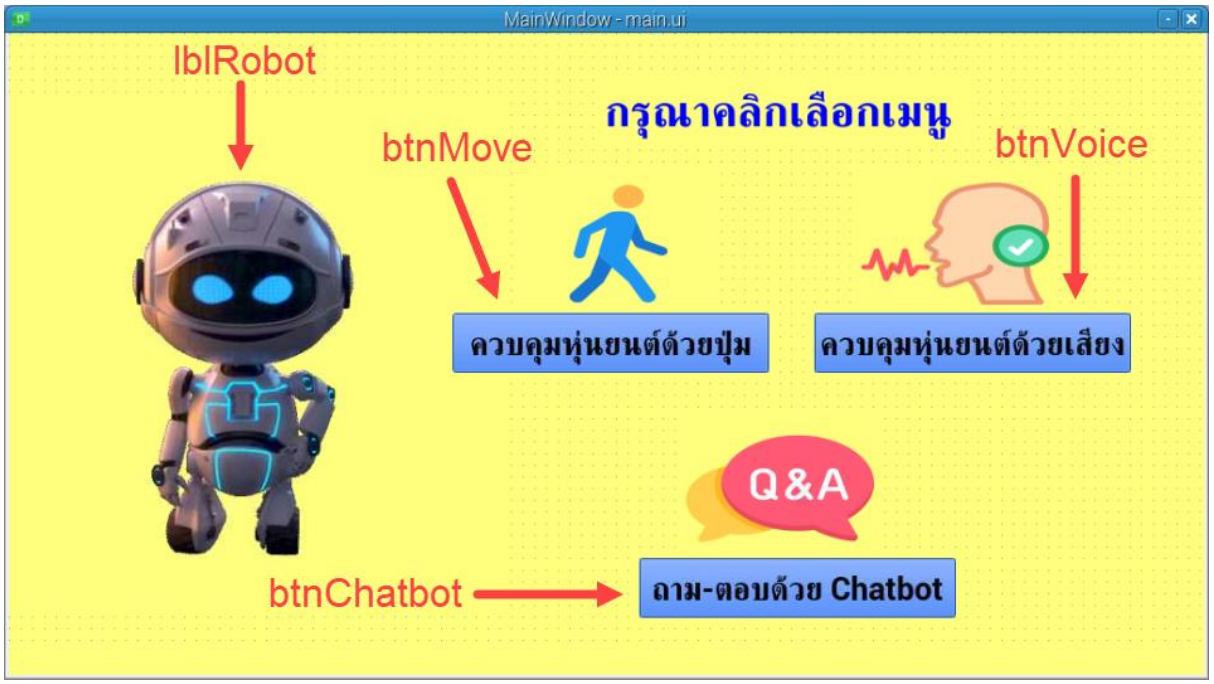
2.9 สร้างปุ่มเมนู “ถาม-ตอบด้วย Chatbot” โดยคัดลอกปุ่มใดก็ได้ จากนั้นแก้ไขข้อความบนปุ่มเป็น “ถาม-ตอบด้วย Chatbot” แล้วเปลี่ยนชื่อปุ่ม โดยคลิกที่ objectName ตั้งชื่อเป็น “btnChatbot”



2.10 ใส่รูปภาพประกอบปุ่มคำสั่งทั้ง 3 ปุ่ม

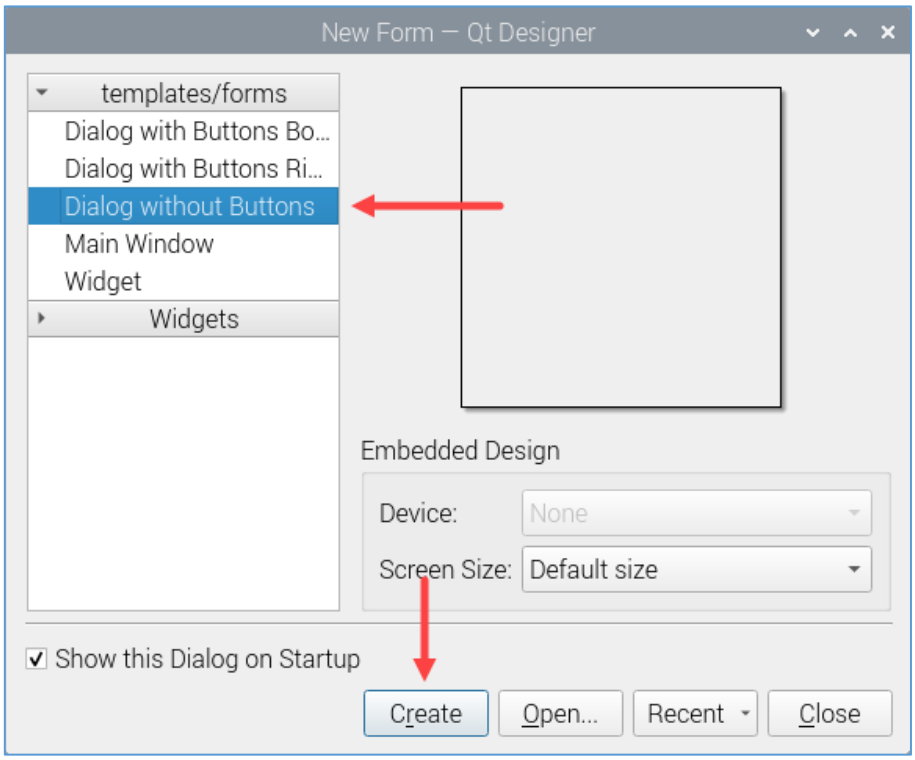
1) คัดลอก Label รูปภาพหุ่นยนต์ จากนั้นแก้ไขรูปภาพในช่อง pixmap ให้สอดคล้องกับปุ่ม คัดลอกและทำให้ครบทั้ง 3 รูปภาพ

ตั้งชื่อ object แต่ละอันในช่อง objectName ดังรูป

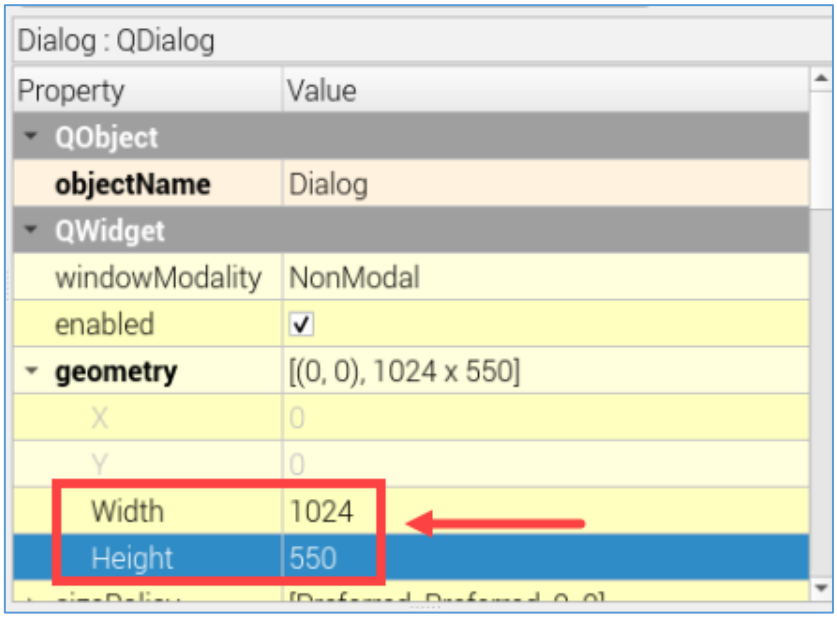


3. ออกแบบหน้าจอ “ควบคุมหุ่นยนต์ด้วยปุ่ม” (move.ui)

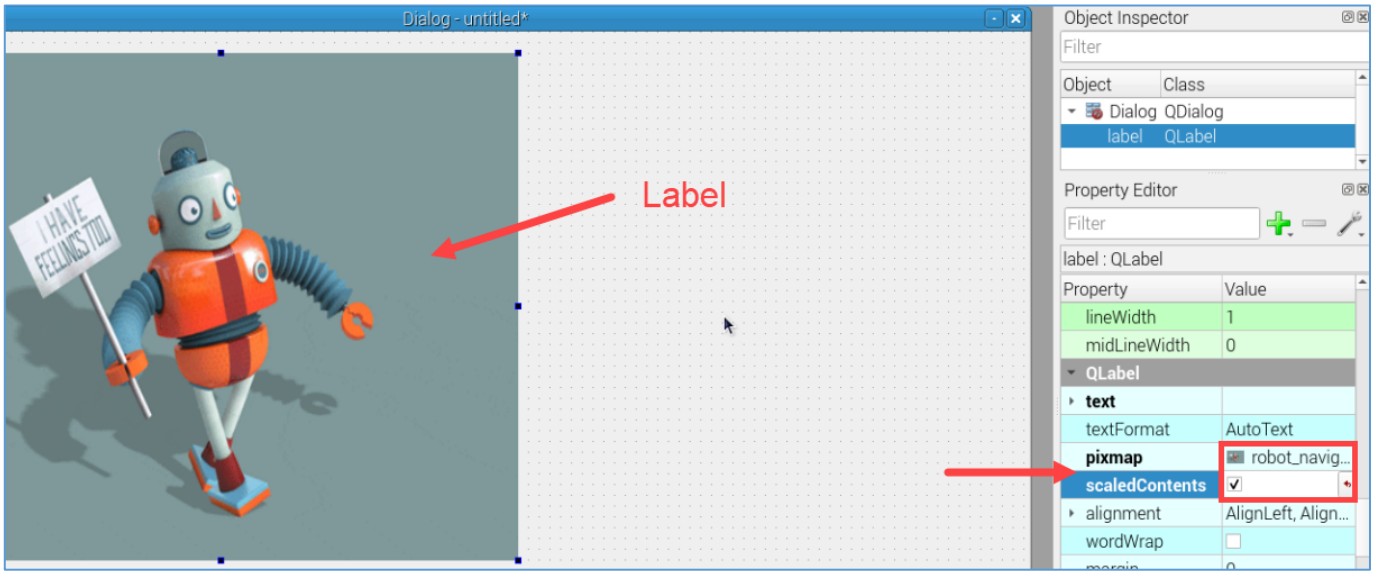
3.1 คลิกแท็บ File --> New จากนั้นเลือกเท็มเพลต Dialog without Buttons แล้วคลิกปุ่ม Create



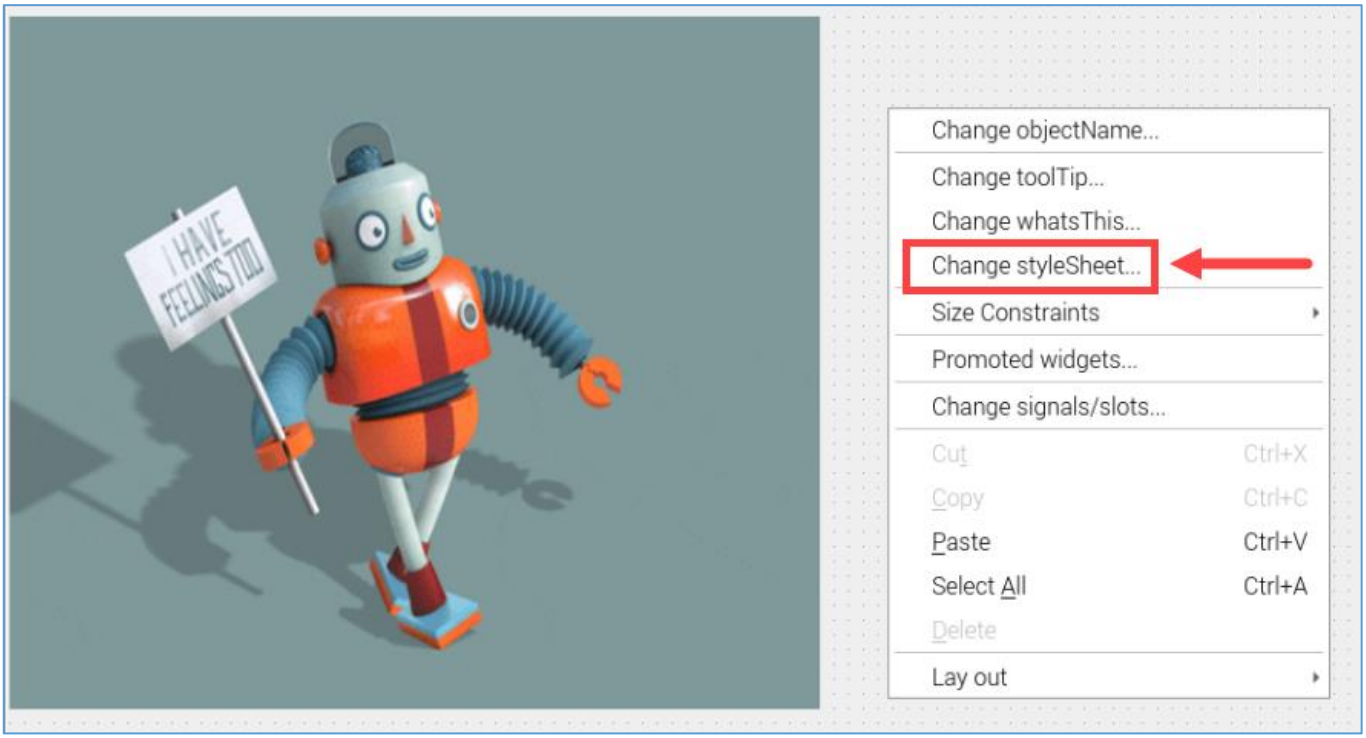
3.2 **กำหนดขนาดหน้าจอ** โดยใส่ค่า Width: 1024 และ Height: 550



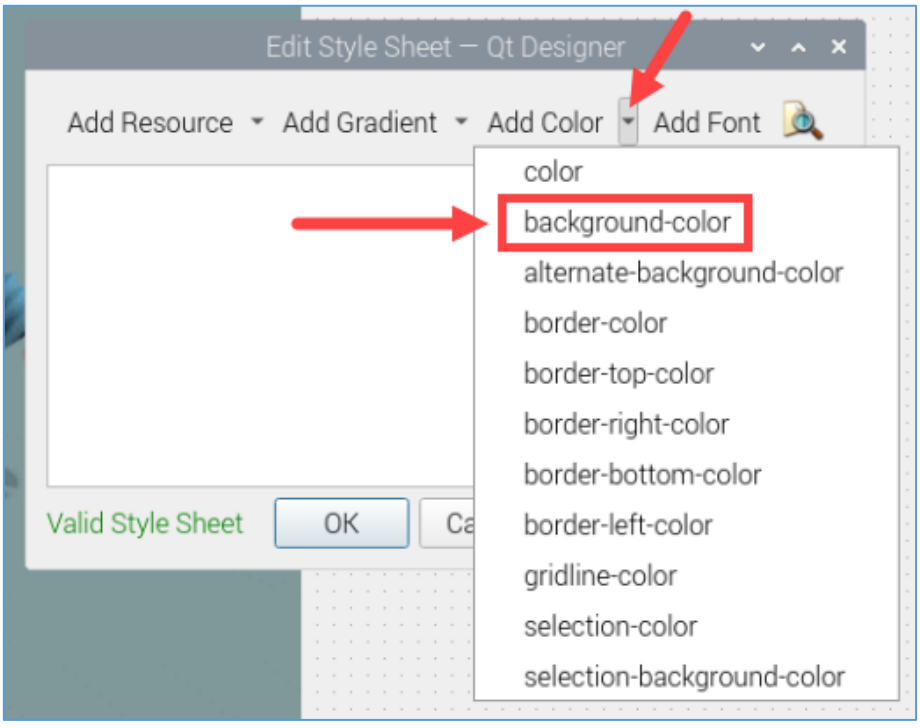
3.3 **แทรกรูปภาพเคลื่อนไหว .gif** โดยลาก Label มาวาง แล้วคลิกที่ pixmap เลือกรูปภาพ .gif แล้วคลิกที่ scaledContents เพื่อปรับรูปภาพให้พอดีกับ Label



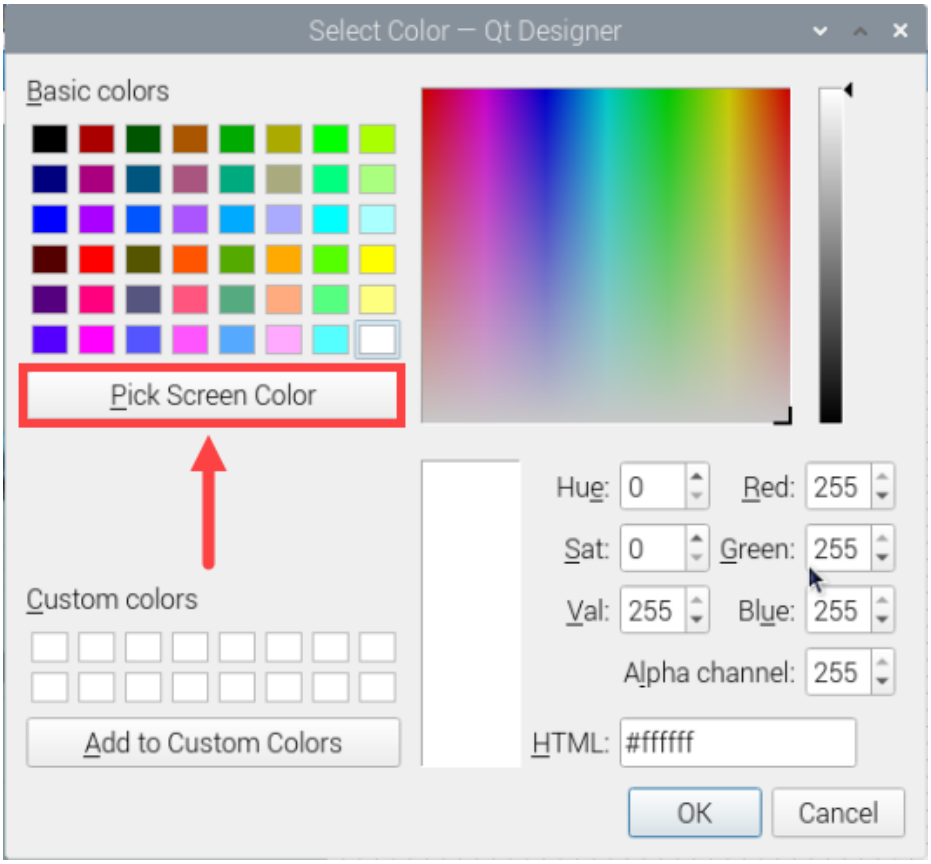
3.4 **ใส่สีพื้นหลังให้เป็นสีเดียวกับพื้นหลังรูปภาพ** โดยคลิกขวาที่หน้าจอ Dialog แล้วเลือก Change styleSheet



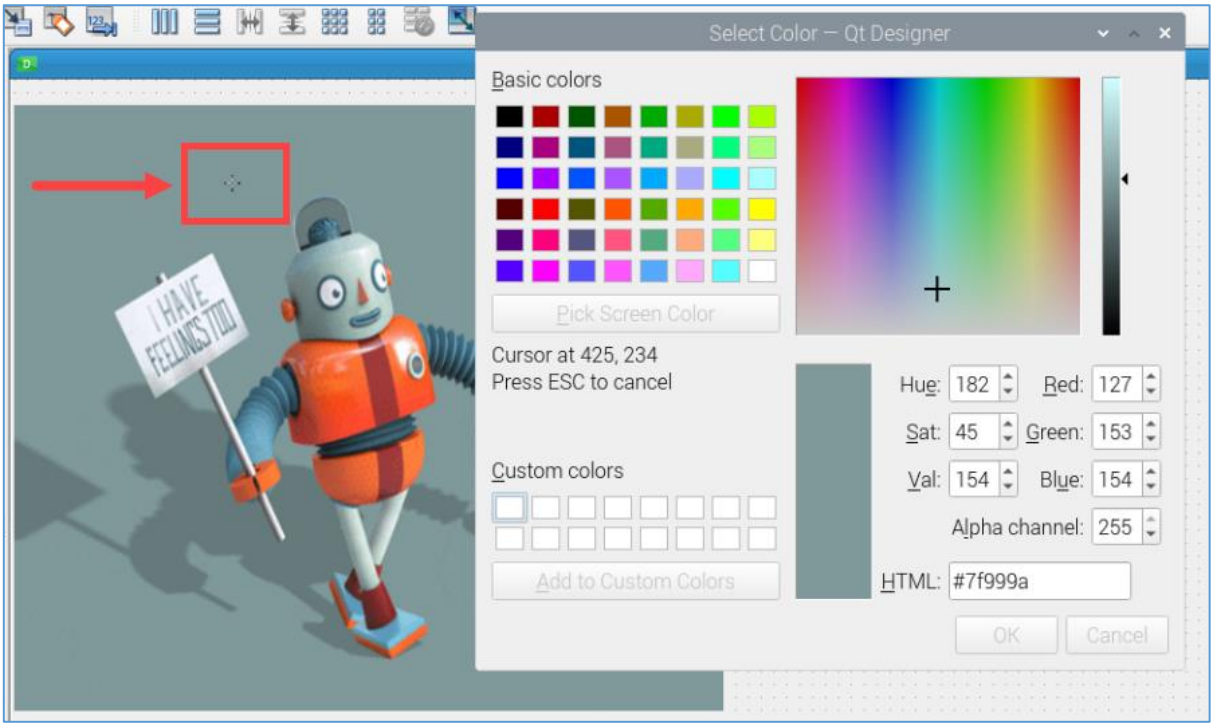
1) คลิกลูกศรด้านหลัง Add Color จากนั้นเลือก background-color



2) คลิกปุ่ม Pick Screen Color

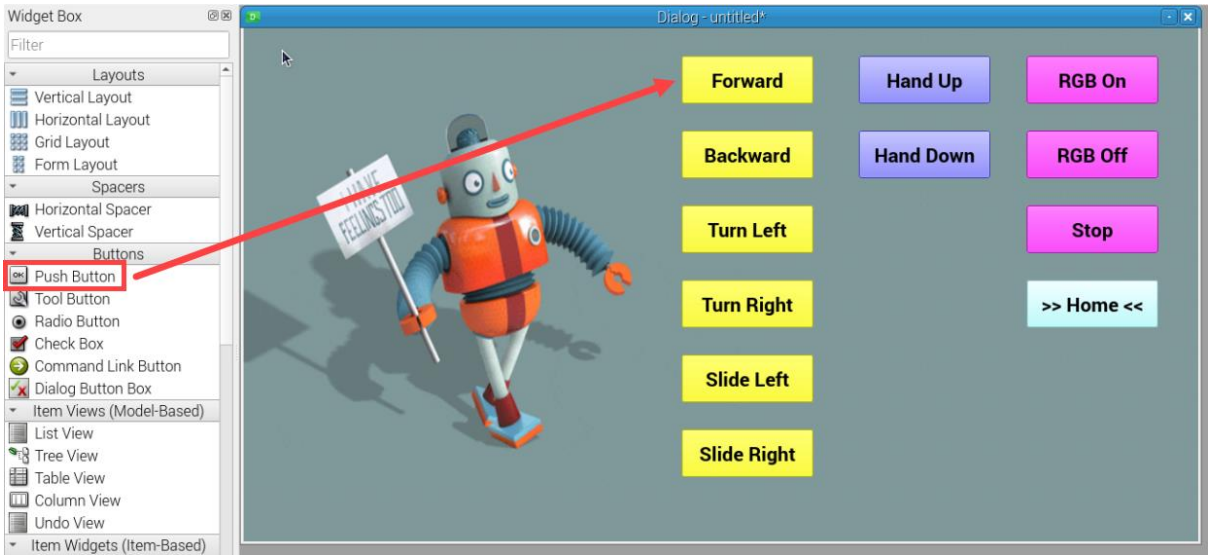


3) นำเมาส์ไปคลิกตรงพื้นหลังรูปภาพ แล้วคลิกปุ่ม OK

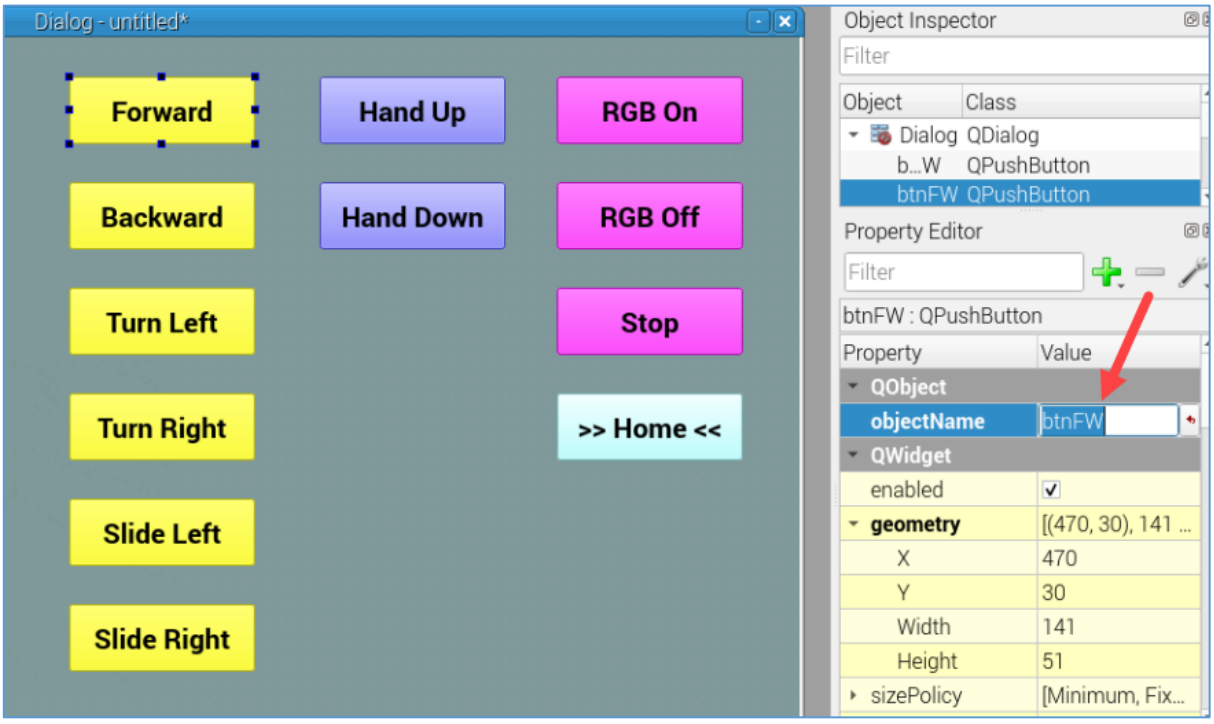


3.5 **แทรกปุ่ม**

1) ลาก Push Button มาวาง แล้วเปลี่ยนข้อความบนปุ่มและเปลี่ยนสีพื้นหลังปุ่ม ดังนี้



2) กำหนด Property: objectName ของแต่ละปุ่ม ดังนี้



- ปุ่ม Forward --> btnFW

- ปุ่ม Backward --> btnBW

- ปุ่ม Turn Left --> btnTL

- ปุ่ม Turn Right --> btnTR

- ปุ่ม Slide Left --> btnSL

- ปุ่ม Slide Right --> btnSR

- ปุ่ม Hand Up --> btnHU

- ปุ่ม Hand Down --> btnHD

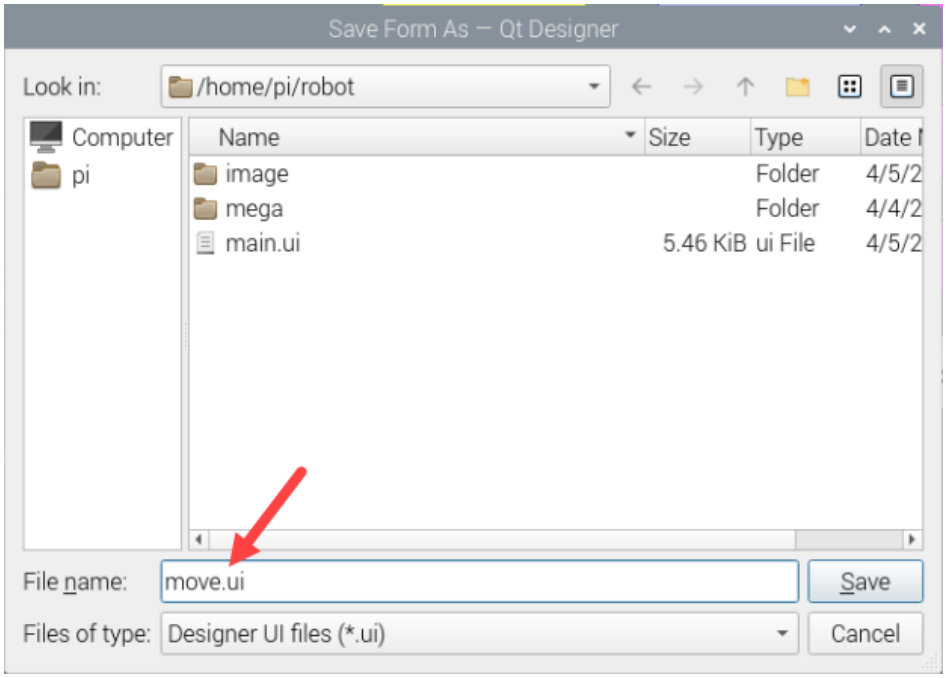
- ปุ่ม RGB On --> btnRGB1

- ปุ่ม RGB Off --> btnRGB0

- ปุ่ม Stop --> btnStop

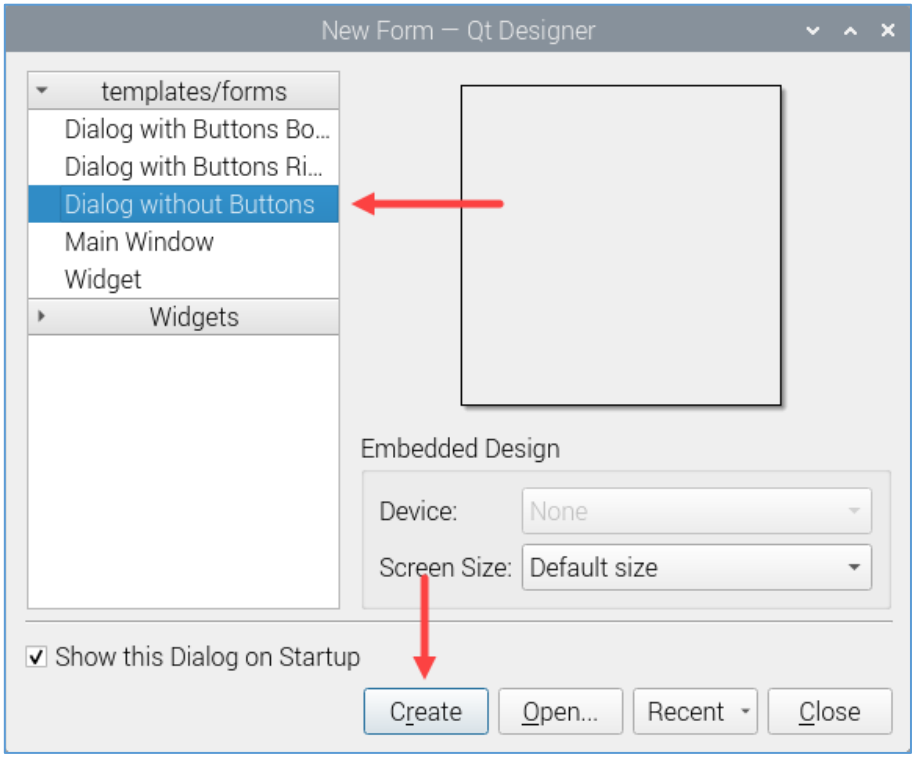
- ปุ่ม Home --> btnHome

3.6 บันทึกไฟล์ในโฟลเดอร์ robot โดยตั้งชื่อเป็น move.ui

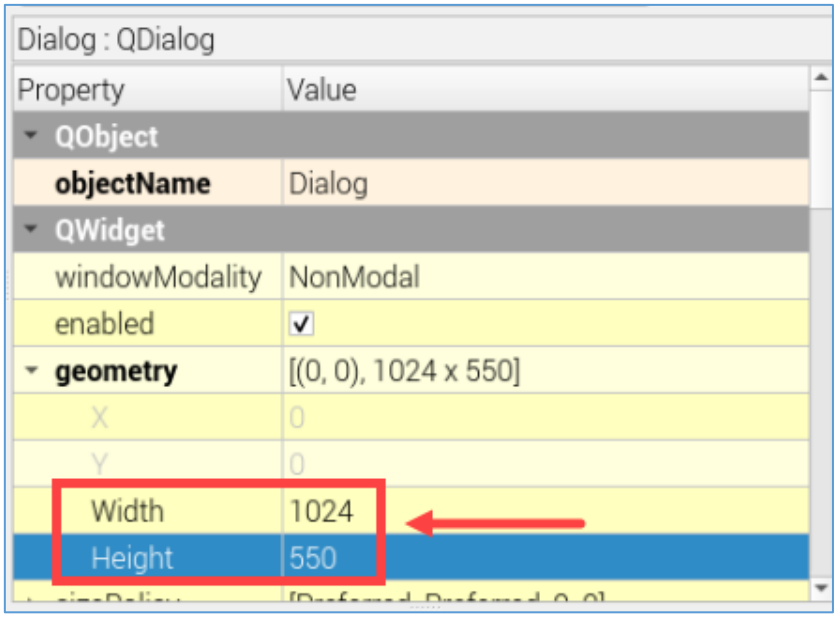


4. ออกแบบหน้าจอ “ควบคุมหุ่นยนต์ด้วยเสียง” (voicecontrol.ui)

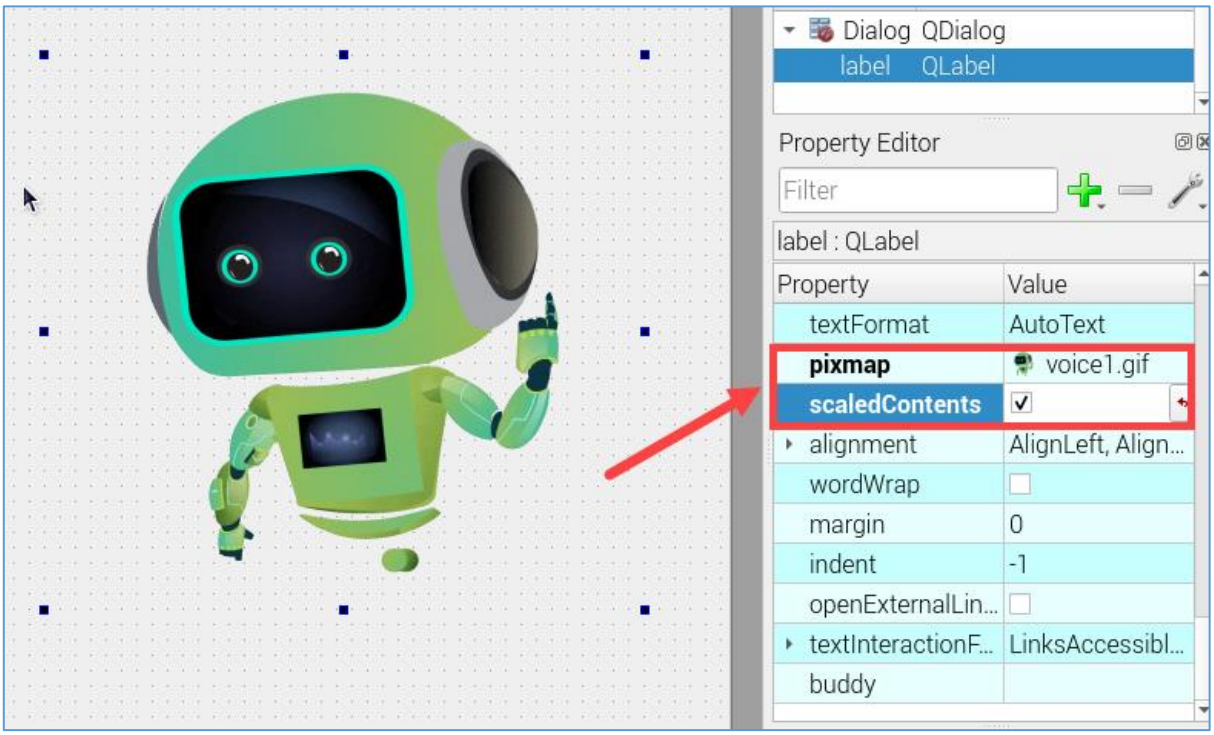
4.1 คลิกแท็บ File --> New จากนั้นเลือกเท็มเพลต Dialog without Buttons แล้วคลิกปุ่ม Create



4.2 **กำหนดขนาดหน้าจอ** โดยใส่ค่า Width: 1024 และ Height: 550



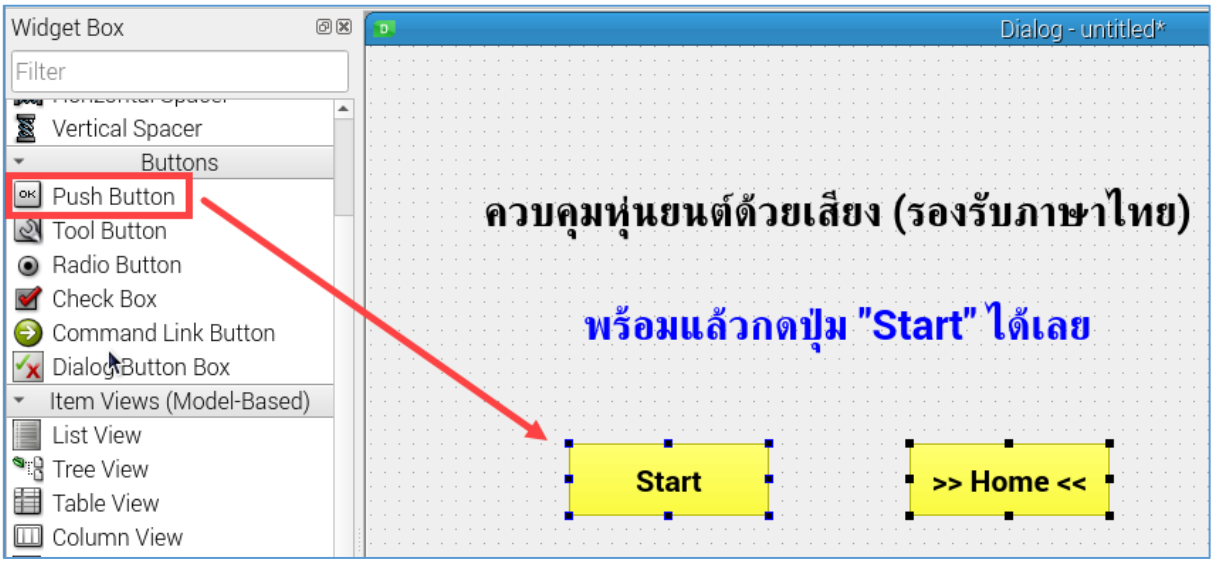
4.3 **แทรกรูปภาพเคลื่อนไหว .gif** โดยลาก Label มาวาง แล้วคลิกที่ pixmap เลือกรูปภาพ .gif แล้วคลิกที่ scaledContents เพื่อปรับรูปภาพให้พอดีกับ Label



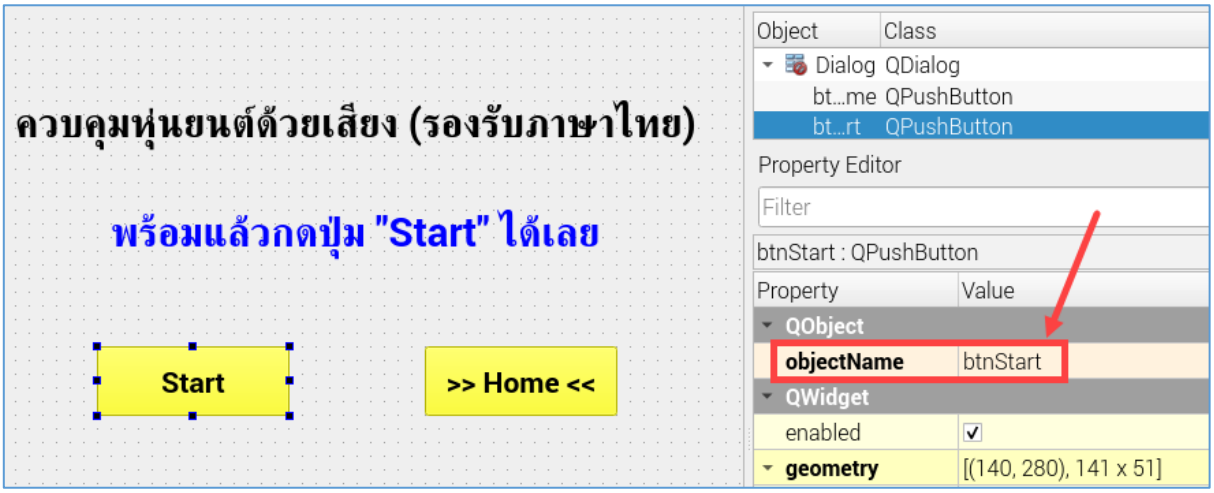
4.4 **แทรกข้อความ** โดยลาก Label มาวางแล้วแก้ไขข้อความและปรับ font

4.5 **แทรกปุ่ม Start และ Home**

1) ลาก Push Button มาวาง แล้วเปลี่ยนข้อความบนปุ่ม ดังนี้



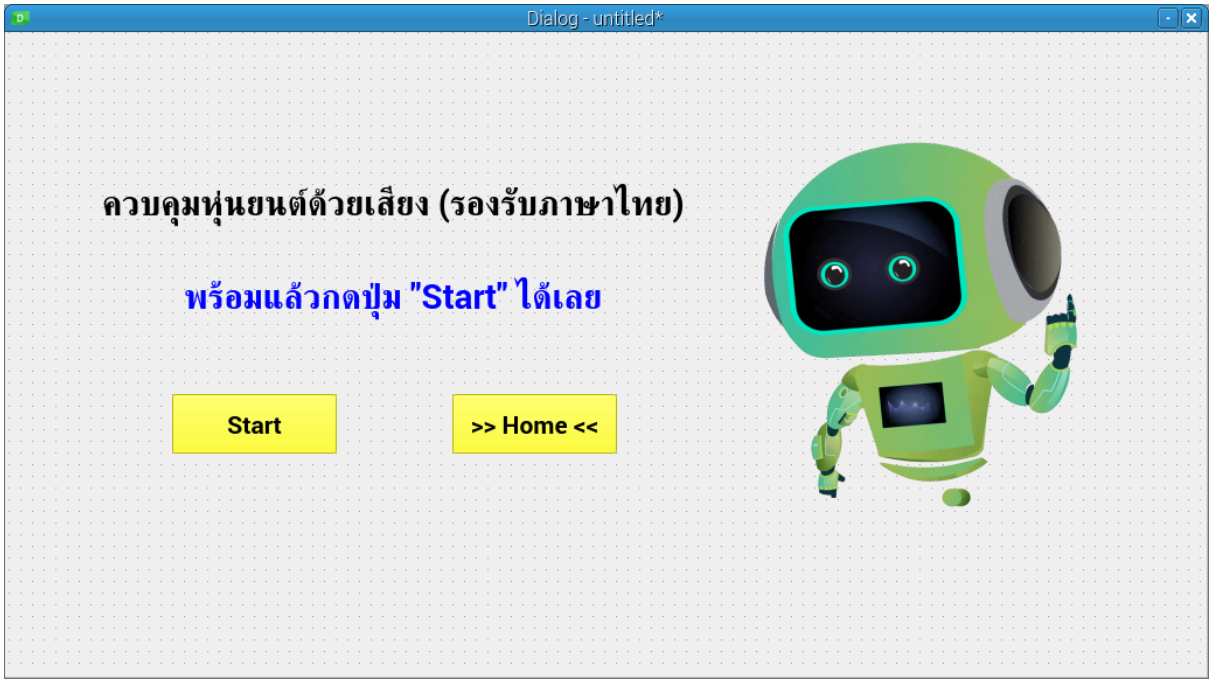
2) กำหนด Property: objectName ของปุ่ม ดังนี้



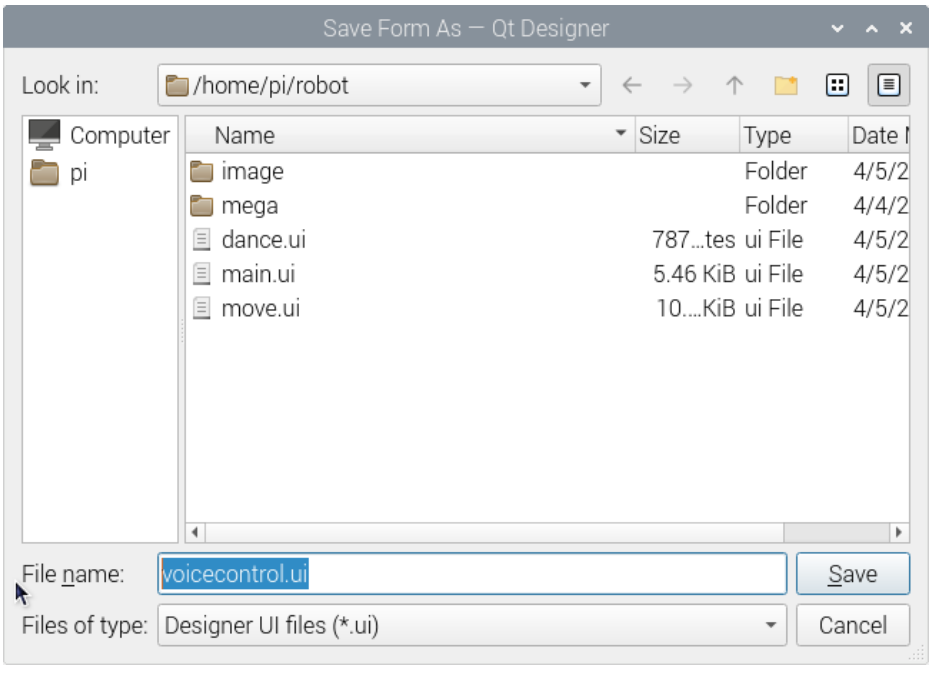
- ปุ่ม Start --> btnStart

- ปุ่ม Start --> btnHome

จะได้ผลลัพธ์ ดังรูป

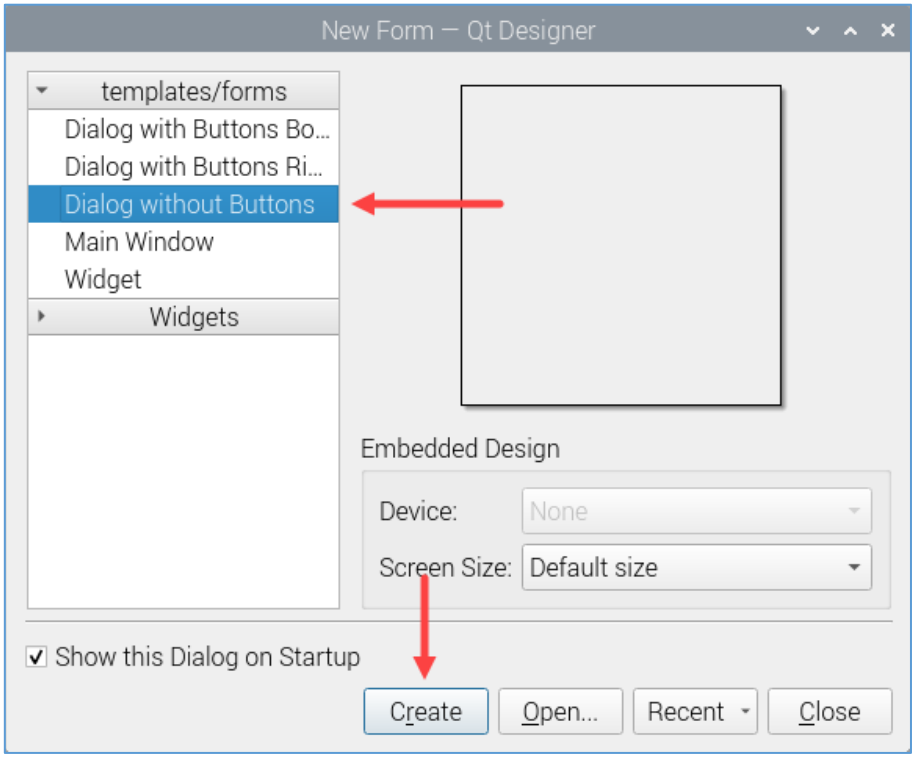


4.6 บันทึกไฟล์ในโฟลเดอร์ robot โดยตั้งชื่อเป็น voicecontrol.ui

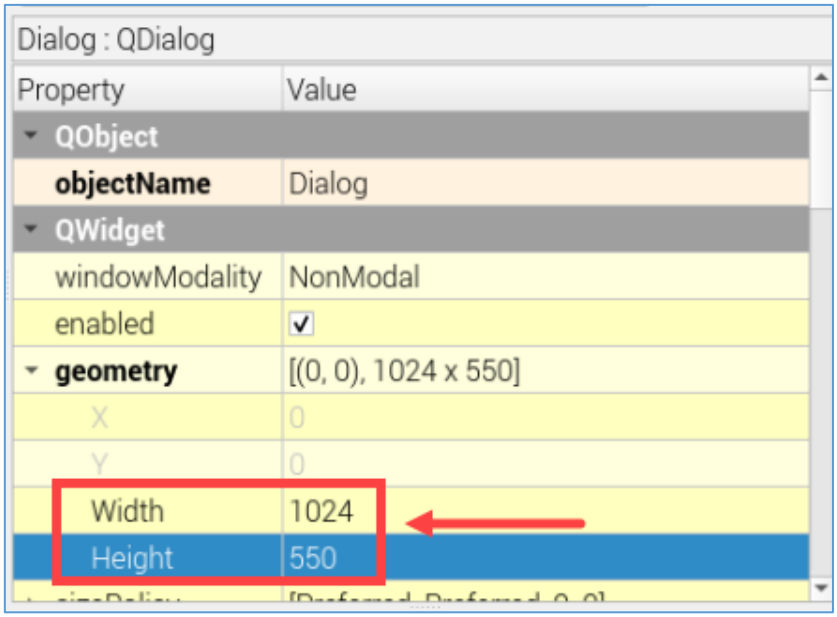


5. ออกแบบหน้าจอ “ถาม-ตอบด้วย Chatbot” (chatbot.ui)

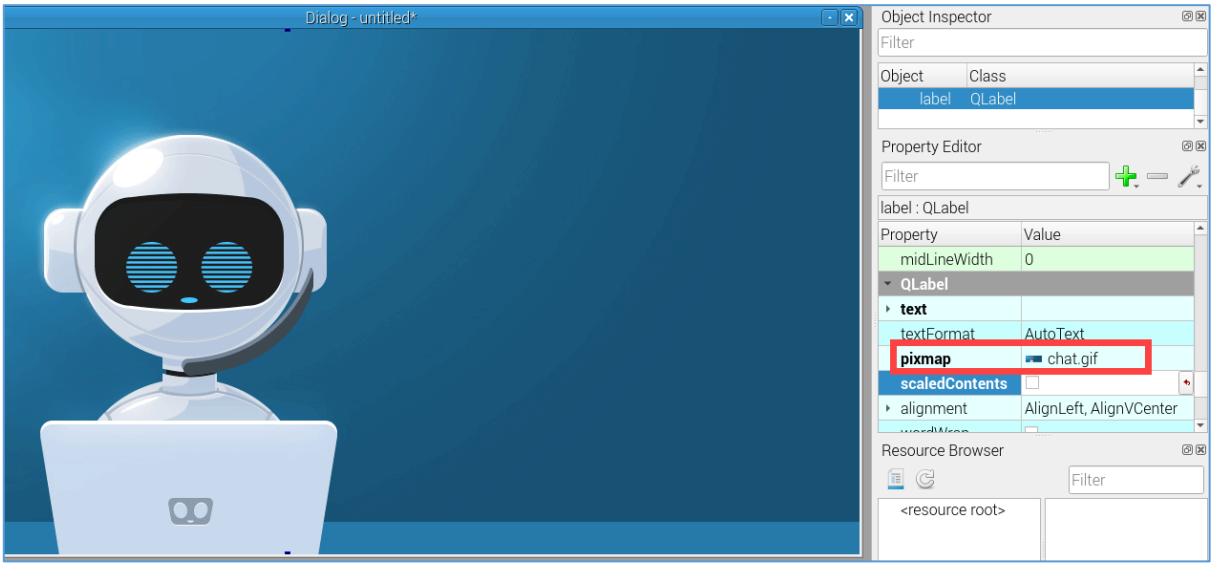
5.1 คลิกแท็บ File --> New จากนั้นเลือกเท็มเพลต Dialog without Buttons แล้วคลิกปุ่ม Create



5.2 **กำหนดขนาดหน้าจอ** โดยใส่ค่า Width: 1024 และ Height: 550

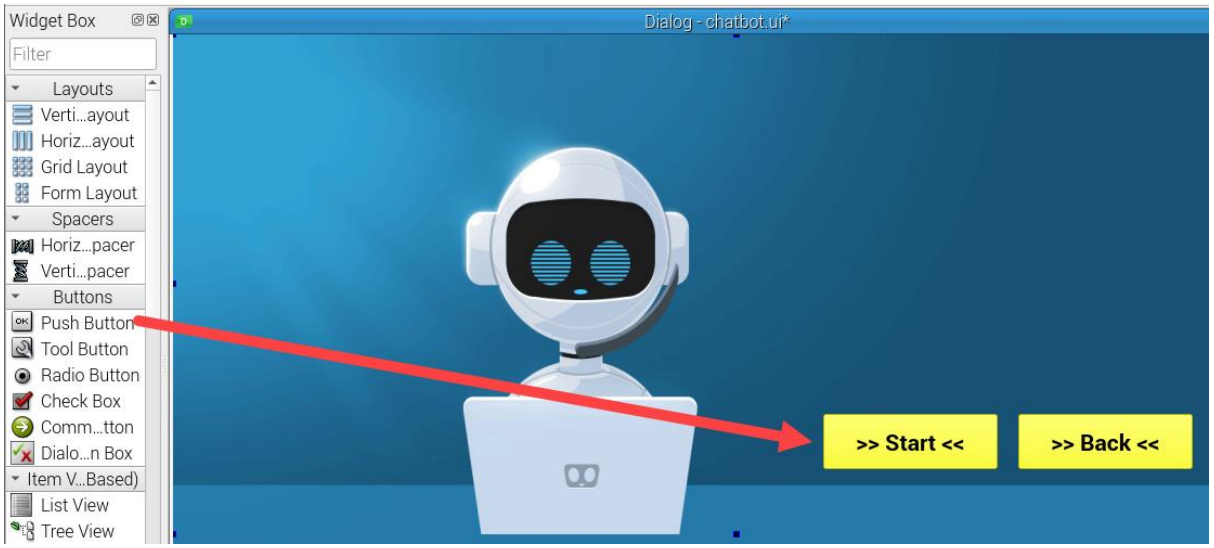


5.3 **แทรกรูปภาพเคลื่อนไหว .gif** โดยลาก Label มาวาง แล้วคลิกที่ pixmap เลือกรูปภาพ .gif

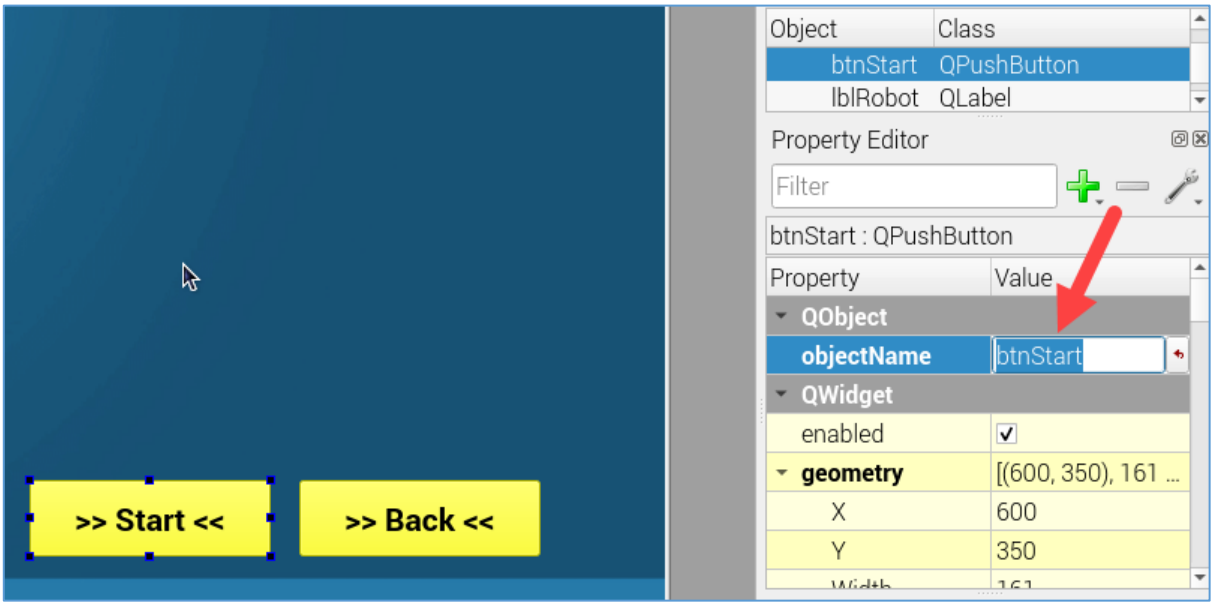


5.4 **แทรกปุ่ม Start และ Home**

1) ลาก Push Button มาวาง แล้วเปลี่ยนข้อความบนปุ่ม ดังนี้



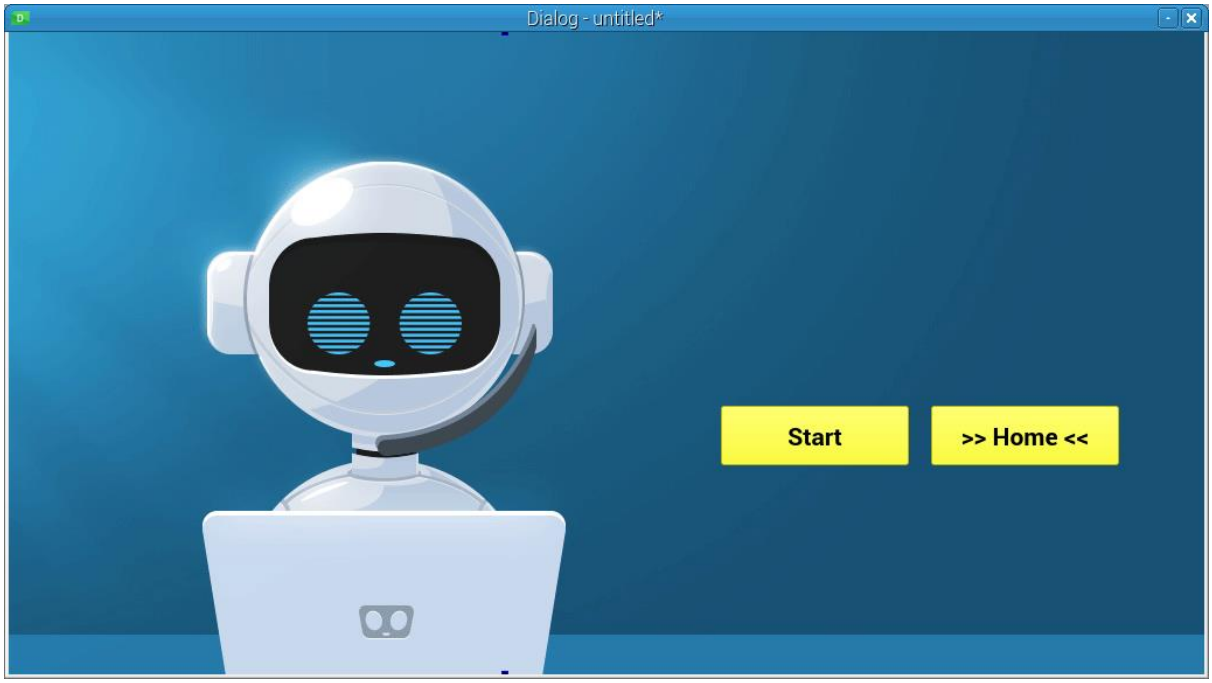
2) กำหนด Property: objectName ของปุ่ม ดังนี้



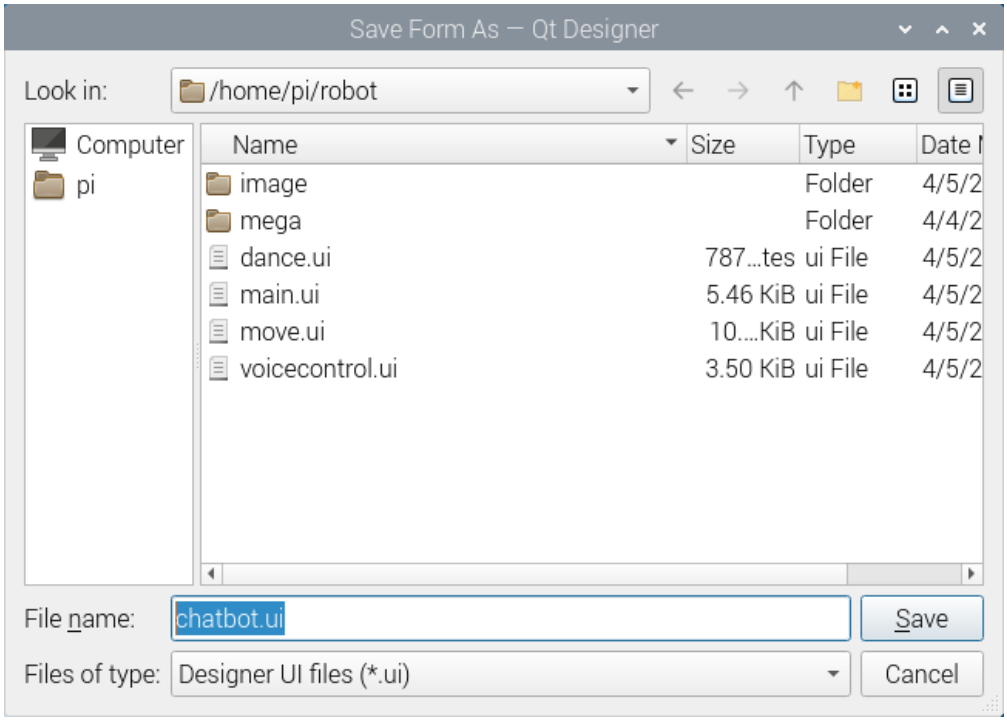
- ปุ่ม Start --> btnStart

- ปุ่ม Start --> btnHome

จะได้ผลลัพธ์ ดังรูป



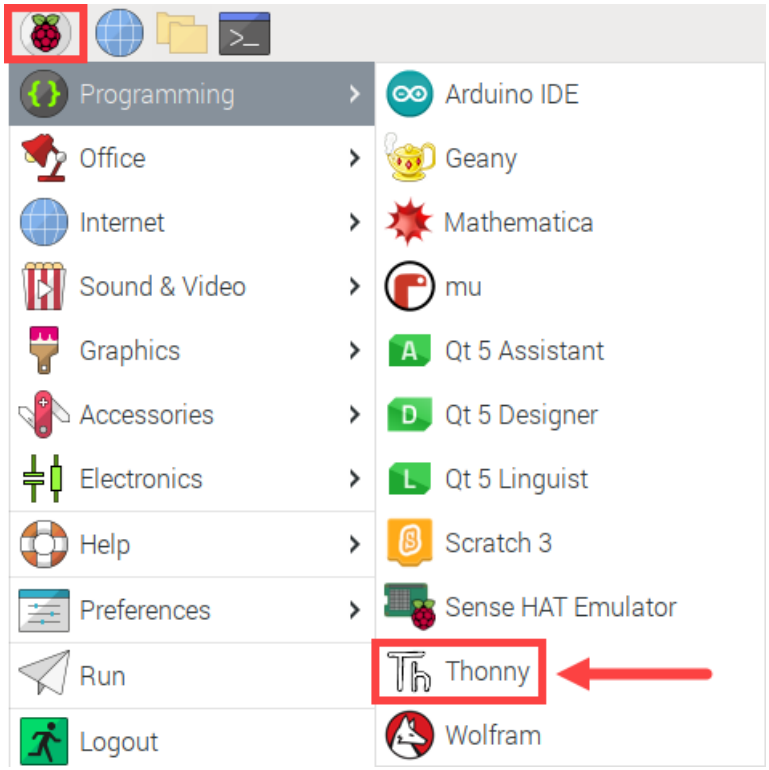
5.5 บันทึกไฟล์ในโฟลเดอร์ robot โดยตั้งชื่อเป็น chatbot.ui



เขียนโปรแกรมภาษาไพธอนด้วยบอร์ด Raspberry Pi

1. เขียนคำสั่งแสดงหน้าจอ GUI และสร้างการเชื่อมโยงแต่ละหน้าจอ

1.1 **เปิดโปรแกรม Thonny Python IDE** เพื่อเขียนโปรแกรม โดยคลิกที่ Pi --> Programming --> Thonny



1.2 เขียนโปรแกรมเพื่อให้แสดงหน้าจอ GUI และสร้างลิงก์เชื่อมโยงระหว่างหน้าจอ ดังนี้

```
import sys, os
from PyQt5.QtGui import *
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.uic import *

class MainWindow(QMainWindow): #หน้าต่างเมนูหลัก
    def __init__(self): #ฟังก์ชันหลักของหน้าต่าง GUI
        super(MainWindow, self).__init__()
        loadUi('main.ui',self) #เรียกหน้าต่างออกแบบขึ้นมา
        #Button
        self.btnVoice.clicked.connect(self.Voicecontrol)
        self.btnChatbot.clicked.connect(self.Chatbot)
        self.btnMove.clicked.connect(self.Move)

    @pyqtSlot()
    def Voicecontrol(self):
        widget.addWidget(Voicecontrol())
        widget.setFixedHeight(550)
        widget.setFixedWidth(1024)
        widget.setCurrentIndex(widget.currentIndex()+1)

    def Chatbot(self):
        widget.addWidget(Chatbot())
        widget.setFixedHeight(550)
        widget.setFixedWidth(1024)
        widget.setCurrentIndex(widget.currentIndex()+1)

    def Move(self):
        widget.addWidget(Move())
        widget.setFixedHeight(550)
        widget.setFixedWidth(1024)
        widget.setCurrentIndex(widget.currentIndex()+1)

class Voicecontrol(QDialog): #สั่งการด้วยเสียง
    def __init__(self):
        super(Voicecontrol, self).__init__()
        loadUi("voicecontrol.ui", self)
        self.btnStart.clicked.connect(self.Start)
        self.btnHome.clicked.connect(self.Home)

    @pyqtSlot()
    def Start(self):
        print("Start")

    def Home(self):
        widget.addWidget(MainWindow())
        widget.setFixedHeight(550)
        widget.setFixedWidth(1024)
        widget.setCurrentIndex(widget.currentIndex()+1)

```

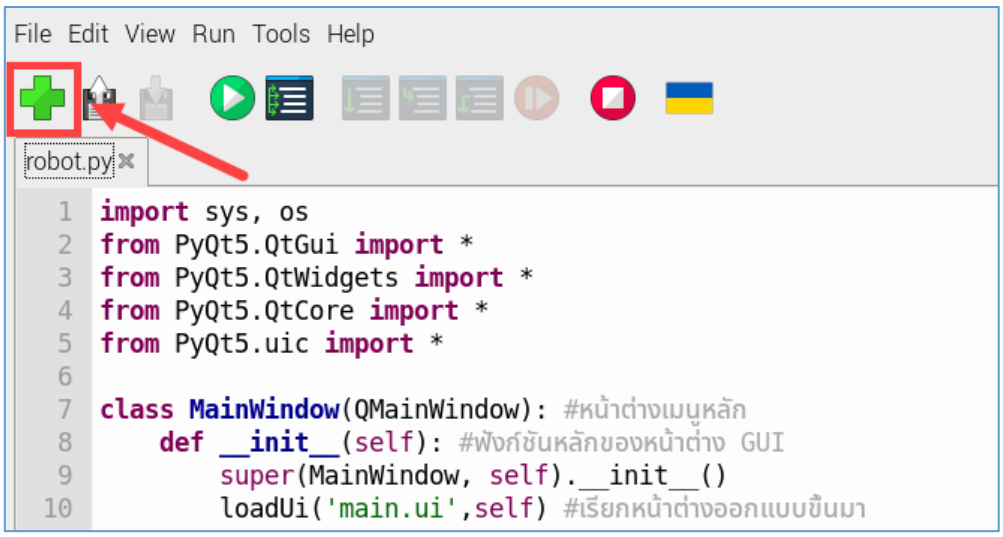
```
class Chatbot(QDialog): #ถาม-ตอบด้วยแชทบอต
    def __init__(self):
        super(Chatbot, self).__init__()
        loadUi("chatbot.ui", self)
        self.btnStart.clicked.connect(self.Chat)
        self.btnHome.clicked.connect(self.Home)

    @pyqtSlot()
    def Home(self):
        widget.addWidget(MainWindow())
        widget.setFixedHeight(550)
        widget.setFixedWidth(1024)
        widget.setCurrentIndex(widget.currentIndex()+1)

    def Chat(self):
        print("Chat")

class Move(QDialog): #กดปุ่มเพื่อควบคุมหุ่นยนต์
    def __init__(self):
        super(Move, self).__init__()
        loadUi("move.ui", self)
        self.btnHome.clicked.connect(self.Home)
        self.btnFW.clicked.connect(self.FW)
        self.btnBW.clicked.connect(self.BW)
        self.btnTR.clicked.connect(self.TR)
        self.btnTL.clicked.connect(self.TL)
        self.btnSL.clicked.connect(self.SL)
        self.btnSR.clicked.connect(self.SR)
        self.btnHU.clicked.connect(self.HU)
        self.btnHD.clicked.connect(self.HD)
        self.btnRGB0.clicked.connect(self.RGB0)
        self.btnRGB1.clicked.connect(self.RGB1)
        self.btnStop.clicked.connect(self.Stopall)

    @pyqtSlot()
    def Home(self):
        widget.addWidget(MainWindow())
        widget.setFixedHeight(550)
        widget.setFixedWidth(1024)
        widget.setCurrentIndex(widget.currentIndex()+1)

    def FW(self):
        print("Forward")

    def BW(self):
        print("Backward")

    def TL(self):
        print("Turn Left")

```

```
102     def TR(self):
103         print("Turn Right")
104 
105     def SR(self):
106         print("Slide Right")
107 
108     def SL(self):
109         print("Slide Left")
110 
111     def HU(self):
112         print("Hand Up")
113 
114     def HD(self):
115         print("Hand Down")
116 
117     def RGB0(self):
118         print("RGB Off")
119 
120     def RGB1(self):
121         print("RGB On")
122 
123     def Stopall(self):
124         print("Stop All")
125 
126 app=QApplication(sys.argv) #เริ่มต้นสร้าง GUI
127 widget = QStackedWidget() #สร้างหน้าต่าง GUI
128 widget.addWidget(MainWindow()) #เรียกหน้าต่างขึ้นมาก่อน
129 widget.setFixedHeight(550) #กำหนดความสูงหน้าจอ
130 widget.setFixedWidth(1024) #กำหนดความกว้างหน้าจอ
131 widget.show() #แสดงหน้าจอ
132 sys.exit(app.exec_())
```

2. เขียนโปรแกรมให้เล่นเพลงเมื่อเปิดหน้าหลัก

2.1 ทดสอบเขียนโปรแกรม

1) เปิดไฟล์ใหม่ขึ้นมา โดยคลิกที่ New



2) เขียนโค้ดคำสั่งให้เล่นเสียง จากนั้นกดปุ่ม Run

```python
import pygame
import time

pygame.mixer.init()                     # กำหนดให้ pygame mixer ใช้ระบบเสียง
pygame.mixer.music.load('song.mp3')     # โหลดไฟล์ MP3
pygame.mixer.music.set_volume(0.3)      # ระดับเสียง (0.0 = ไม่มีเสียง, 1.0 = เสียงดังสุด)
pygame.mixer.music.play()               # เริ่มเล่นเพลง

while pygame.mixer.music.get_busy():    # รอจนกว่าเพลงจะเล่นเสร็จ
    time.sleep(1)
```

2.2 คัดลอกโค้ดคำสั่งเล่นเสียงไปวางในไฟล์ robot.py ดังรูป

```python
import sys, os
from PyQt5.QtGui import *
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.uic import *
import pygame

pygame.mixer.init()   # กำหนดให้ pygame mixer ใช้ระบบเสียง

class MainWindow(QMainWindow): #หน้าต่างเมนูหลัก
    def __init__(self): #ฟังก์ชันหลักของหน้าต่าง GUI
        super(MainWindow, self).__init__()
        loadUi('main.ui',self) #เรียกหน้าต่างออกแบบขึ้นมา
        #Button
        self.btnVoice.clicked.connect(self.Voicecontrol)
        self.btnChatbot.clicked.connect(self.Chatbot)
        self.btnMove.clicked.connect(self.Move)

        #เล่นเพลง
        pygame.mixer.music.load('song.mp3')     # โหลดไฟล์ MP3
        pygame.mixer.music.set_volume(0.3)      # ระดับเสียง (0.0 = ไม่มีเสียง, 1.0 = เสียงดังสุด)
        pygame.mixer.music.play()               # เริ่มเล่นเพลง
```

2.3 **เขียนโค้ดคำสั่งให้เมื่อคลิกไปหน้าจออื่นให้หยุดเล่นเพลง** โดยพิมพ์คำสั่งในทุกๆ ฟังก์ชันของคลาส MainWindow

```
class MainWindow(QMainWindow): #หน้าต่างเมนูหลัก
    def __init__(self): #ฟังก์ชันหลักของหน้าต่าง GUI
        super(MainWindow, self).__init__()
        loadUi('main.ui',self) #เรียกหน้าต่างออกแบบขึ้นมา
        #Button
        self.btnVoice.clicked.connect(self.Voicecontrol)
        self.btnChatbot.clicked.connect(self.Chatbot)
        self.btnMove.clicked.connect(self.Move)

        #เล่นเพลง
        pygame.mixer.music.load('song.mp3')      # โหลดไฟล์ MP3
        pygame.mixer.music.set_volume(0.3)       # ระดับเสียง (0.0 = ไม่มีเสียง, 1.0 = เสียงดังสุด)
        pygame.mixer.music.play()                # เริ่มเล่นเพลง

    @pyqtSlot()
    def Voicecontrol(self):
        pygame.mixer.music.stop()
        widget.addWidget(Voicecontrol())
        widget.setFixedHeight(550)
        widget.setFixedWidth(1024)
        widget.setCurrentIndex(widget.currentIndex()+1)

    def Chatbot(self):
        pygame.mixer.music.stop()
        widget.addWidget(Chatbot())
        widget.setFixedHeight(550)
        widget.setFixedWidth(1024)
        widget.setCurrentIndex(widget.currentIndex()+1)

    def Move(self):
        pygame.mixer.music.stop()
        widget.addWidget(Move())
        widget.setFixedHeight(550)
        widget.setFixedWidth(1024)
        widget.setCurrentIndex(widget.currentIndex()+1)
```

3. เขียนโปรแกรมให้รูปภาพ .gif เล่นภาพเคลื่อนไหว

3.1 เพิ่มโค้ดคำสั่งในคลาส MainWindow() ดังนี้

```
11 class MainWindow(QMainWindow): #หน้าต่างเมนูหลัก
12     def __init__(self): #ฟังก์ชันหลักของหน้าต่าง GUI
13         super(MainWindow, self).__init__()
14         loadUi('main.ui',self) #เรียกหน้าต่างออกแบบขึ้นมา
15         #Button
16         self.btnVoice.clicked.connect(self.Voicecontrol)
17         self.btnChatbot.clicked.connect(self.Chatbot)
18         self.btnDance.clicked.connect(self.Dance)
19         self.btnMove.clicked.connect(self.Move)
20 
21         #รูปภาพ .gif เคลื่อนไหว
22         self.movie = QMovie("image/robot_hello.gif")
23         self.lblRobot.setMovie(self.movie)
24         self.movie.start()
25 
26         #เล่นเพลง
27         pygame.mixer.music.load('song.mp3')      # โหลดไฟล์
28         pygame.mixer.music.set_volume(0.3)       # ระดับเสียง
29         pygame.mixer.music.play()                # เริ่มเล่นเพล
```

3.2 เพิ่มโค้ดคำสั่งในคลาส Voicecontrol() ดังนี้

```
61 class Voicecontrol(QDialog): #สั่งการด้วยเสียง
62     def __init__(self):
63         super(Voicecontrol, self).__init__()
64         loadUi("voicecontrol.ui", self)
65         self.btnStart.clicked.connect(self.Start)
66         self.btnHome.clicked.connect(self.Home)
67 
68         #รูปภาพ .gif เคลื่อนไหว
69         self.movie = QMovie("image/voice1.gif")
70         self.label.setMovie(self.movie)
71         self.movie.start()
```

3.3 เพิ่มโค้ดคำสั่งในคลาส Chatbot() ดังนี้

```
83 class Chatbot(QDialog): #ถาม-ตอบด้วยแชทบอต
84     def __init__(self):
85         super(Chatbot, self).__init__()
86         loadUi("chatbot.ui", self)
87         self.btnStart.clicked.connect(self.Chat)
88         self.btnHome.clicked.connect(self.Home)
89 
90         #รูปภาพ .gif เคลื่อนไหว
91         self.movie = QMovie("image/chat.gif")
92         self.label.setMovie(self.movie)
93         self.movie.start()
```

3.4 เพิ่มโค้ดคำสั่งในคลาส Move() ดังนี้

```
123 class Move(QDialog): #กดปุ่มเพื่อควบคุมหุ่นยนต์
124     def __init__(self):
125         super(Move, self).__init__()
126         loadUi("move.ui", self)
127         self.btnHome.clicked.connect(self.Home)
128         self.btnFW.clicked.connect(self.FW)
129         self.btnBW.clicked.connect(self.BW)
130         self.btnTR.clicked.connect(self.TR)
131         self.btnTL.clicked.connect(self.TL)
132         self.btnSL.clicked.connect(self.SL)
133         self.btnSR.clicked.connect(self.SR)
134         self.btnHU.clicked.connect(self.HU)
135         self.btnHD.clicked.connect(self.HD)
136         self.btnRGB0.clicked.connect(self.RGB0)
137         self.btnRGB1.clicked.connect(self.RGB1)
138         self.btnStop.clicked.connect(self.Stopall)
139 
140         #รูปภาพ .gif เคลื่อนไหว
141         self.movie = QMovie("image/robot_navigation.gif")
142         self.label.setMovie(self.movie)
143         self.movie.start()
```

4. เขียนคำสั่งสำหรับควบคุมหุ่นยนต์ด้วยปุ่มควบคุม (คลาส Move)

4.1 เชื่อมต่อบอร์ด Arduino Mega 2560 กับ Raspberry Pi

4.2 เขียนโค้ดคำสั่งในการตรวจสอบการเชื่อมต่อกับพอร์ต USB Serial (เขียนไว้ด้านบนคลาส MainWindow)

```
5  from PyQt5.uic import *
6  import pygame
7  import time
8  import serial
9 
10 pygame.mixer.init()                    # กำหนดให้ pygame mixer ใช้ระบบเสียง
11 
12 #ตรวจสอบการเชื่อมต่อกับบอร์ด Arduino
13 if __name__ == '__main__':
14     try:
15         ser = serial.Serial('/dev/ttyUSB0', 9600, timeout = 1)
16         print("USB0")
17     except:
18         ser = serial.Serial('/dev/ttyUSB1', 9600, timeout = 1)
19         print("USB1")
20 
21 class MainWindow(QMainWindow): #หน้าต่างเมนูหลัก
22     def __init__(self): #ฟังก์ชันหลักของหน้าต่าง GUI
23         super(MainWindow, self).__init__()
24         loadUi('main.ui',self) #เรียกหน้าต่างออกแบบขึ้นมา
```

4.3 เขียนโค้ดคำสั่งควบคุมหุ่นยนต์ด้วยปุ่ม โดยใส่โค้ดในแต่ละฟังก์ชันของคลาส Move()

```
133 class Move(QDialog): #กดปุ่มเพื่อควบคุมหุ่นยนต์
134     def __init__(self):
135         super(Move, self).__init__()
136         loadUi("move.ui", self)
137         self.btnHome.clicked.connect(self.Home)
138         self.btnFW.clicked.connect(self.FW)
139         self.btnBW.clicked.connect(self.BW)
140         self.btnTR.clicked.connect(self.TR)
141         self.btnTL.clicked.connect(self.TL)
142         self.btnSL.clicked.connect(self.SL)
143         self.btnSR.clicked.connect(self.SR)
144         self.btnHU.clicked.connect(self.HU)
145         self.btnHD.clicked.connect(self.HD)
146         self.btnRGB0.clicked.connect(self.RGB0)
147         self.btnRGB1.clicked.connect(self.RGB1)
148         self.btnStop.clicked.connect(self.Stopall)
149 
150         #รูปภาพ .gif เคลื่อนไหว
151         self.movie = QMovie("image/robot_navigation.gif")
152         self.label.setMovie(self.movie)
153         self.movie.start()
154 
155     @pyqtSlot()
156     def Home(self):
157         widget.addWidget(MainWindow())
158         widget.setFixedHeight(550)
159         widget.setFixedWidth(1024)
160         widget.setCurrentIndex(widget.currentIndex()+1)
161 
162     def FW(self):
163         print("Forward")
164         ser.write(b"FW\n")
165         time.sleep(1)
166         ser.reset_input_buffer() #ล้างบัฟเฟอร์อินพุตเพื่อเริ่มสื่อสารกับบอร์ด Aduino ใหม่
167 
168     def BW(self):
169         print("Backward")
170         ser.write(b"BW\n")
171         time.sleep(1)
172         ser.reset_input_buffer()
173 
174     def TL(self):
175         print("Turn Left")
176         ser.write(b"TL\n")
177         time.sleep(1)
178         ser.reset_input_buffer()
```

```
179
180     def TR(self):
181         print("Turn Right")
182         ser.write(b"TR\n")
183         time.sleep(1)
184         ser.reset_input_buffer()
185
186     def SR(self):
187         print("Slide Right")
188         ser.write(b"SR\n")
189         time.sleep(1)
190         ser.reset_input_buffer()
191
192     def SL(self):
193         print("Slide Left")
194         ser.write(b"SL\n")
195         time.sleep(1)
196         ser.reset_input_buffer()
197
198     def HU(self):
199         print("Hand Up")
200         ser.write(b"HU\n")
201         time.sleep(1)
202         ser.reset_input_buffer()
203
204     def HD(self):
205         print("Hand Down")
206         ser.write(b"HD\n")
207         time.sleep(1)
208         ser.reset_input_buffer()
209
210     def RGB0(self):
211         print("RGB Off")
212         ser.write(b"RGBoff\n")
213         ser.reset_input_buffer()
214
215     def RGB1(self):
216         print("RGB On")
217         for i in range(2):
218             ser.write(b"Red\n")
219             ser.write(b"Green\n")
220             ser.write(b"Blue\n")
221             ser.reset_input_buffer()
222
```

```
204     def HD(self):
205         print("Hand Down")
206         ser.write(b"HD\n")
207         time.sleep(1)
208         ser.reset_input_buffer()
209 
210     def RGB0(self):
211         print("RGB Off")
212         ser.write(b"RGBoff\n")
213         ser.reset_input_buffer()
214 
215     def RGB1(self):
216         print("RGB On")
217         for i in range(2):
218             ser.write(b"Red\n")
219             ser.write(b"Green\n")
220             ser.write(b"Blue\n")
221             ser.reset_input_buffer()
222 
223     def Stopall(self):
224         print("Stop All")
225         ser.write(b"RGBoff\n")
226         time.sleep(1)
227         ser.write(b"HSTOP\n")
228         time.sleep(1)
229         ser.write(b"STOP\n")
230         time.sleep(1)
231         ser.reset_input_buffer()
```

5. เขียนคำสั่งควบคุมหุ่นยนต์ด้วยเสียง (Speech Recognition)

5.1 ติดตั้งไลบรารี่

1) ติดตั้ง PyAudio (โมดูลสำหรับเล่นและอัดเสียง)

เปิด Terminal แล้วพิมพ์คำสั่ง sudo apt install python3-pyaudio

2) ติดตั้ง Flac

เปิด Terminal แล้วพิมพ์คำสั่ง sudo apt-get install flac

3) ติดตั้ง Speech Recognition

เปิด Terminal แล้วพิมพ์คำสั่ง pip install SpeechRecognition

4) ติดตั้งไลบรารี gTTS (Google Text-to-Speech) (รองรับภาษาไทย)

เปิด Terminal แล้วพิมพ์คำสั่ง pip install gtts

5) ติดตั้งโปรแกรมสำหรับเล่นเสียง mpg321

เปิด Terminal แล้วพิมพ์คำสั่ง sudo apt-get install mpg321

5.2 ตั้งค่าระบบให้สามารถใช้ไมโครโฟนจากกล้องเว็บแคมได้

1) เปิด Terminal จากนั้นพิมพ์คำสั่ง sudo nano /boot/config.txt เพื่อเข้าไปแก้ไขไฟล์

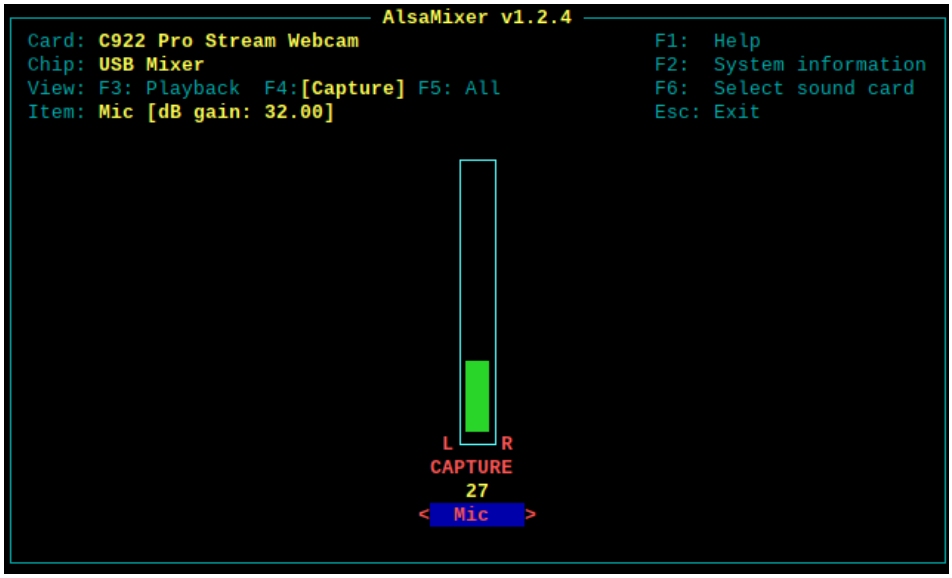
2) หาบรรทัดที่เป็น dtparam=audio=on และเพิ่ม snd-usb-audio ไปที่บรรทัดด้านล่าง

```
dtparam=audio=on
dtoverlay=snd-usb-audio
```

3) บันทึกไฟล์โดยกด Ctrl + X และยืนยันโดยกด Y และ Enter

4) รีสตาร์ท โดยพิมพ์คำสั่ง reboot

5.3 ปรับระดับเสียงของไมโครโฟน โดยเปิด Terminal แล้วพิมพ์ alsamixer จากนั้นกดปุ่ม F6 แล้วเลือกกล้อง Webcam จากนั้นกดปุ่ม F4 ปรับระดับเสียงให้ไม่เกินขีดสีแดง



5.4 ทดลองเขียนโปรแกรม Speech to Text เพื่อรับเสียงแล้วแปลงเป็นข้อความ

```
import speech_recognition as sr
recognizer = sr.Recognizer()
with sr.Microphone() as source:
    print("Say something")
    audio = recognizer.listen(source)
    text = recognizer.recognize_google(audio,language="th-TH")
    print(text)
```

5.5 ทดสอบเขียนโปรแกรม Text to Speech เพื่อแปลงข้อความเป็นเสียงให้หุ่นยนต์พูด

```
from gtts import gTTS
import os

# ข้อความที่ต้องการให้พูด
text = "สวัสดีค่ะ ยินดีที่ได้พบคุณ"

# สร้างเสียงจากข้อความ
tts = gTTS(text=text, lang='th')

# บันทึกเสียงเป็นไฟล์
tts.save("output.mp3")

# เล่นไฟล์เสียง
os.system("mpg321 output.mp3")
```

5.6 เขียนคำสั่งควบคุมหุ่นยนต์ด้วยเสียง โดยใช้ Speech to Text และ Text to Speech

1) นำเข้าไลบรารี

```
import sys, os
from PyQt5.QtGui import *
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5.uic import *
import pygame
import time
import serial
import speech_recognition as sr
from gtts import gTTS

recognizer = sr.Recognizer()
pygame.mixer.init()    # กำหนดให้ pygame mixer ใช้ระบบเสียง
```

2) เขียนโค้ดคำสั่งในคลาส Voice()

```
class Voicecontrol(QDialog): #สั่งการด้วยเสียง
    def __init__(self):
        super(Voicecontrol, self).__init__()
        loadUi("voicecontrol.ui", self)
        self.btnStart.clicked.connect(self.Start)
        self.btnHome.clicked.connect(self.Home)

        #รูปภาพ .gif เคลื่อนไหว
        self.movie = QMovie("image/voice1.gif")
        self.label.setMovie(self.movie)
        self.movie.start()

    @pyqtSlot()
    def Start(self):
        #Text to Speech
        robottalk = "สั่งการหุ่นยนต์ได้เลยค่ะ"
        tts = gTTS(text=robottalk,lang='th')
        tts.save("output.mp3")
        os.system("mpg321 output.mp3")

        while True:
            try:
                with sr.Microphone() as source:
                    print("Say Something")

                    #Speech to Text
                    audio = recognizer.listen(source)
                    talk = recognizer.recognize_google(audio,language="th-TH")

                    if talk == "forward" or talk == "เดิน" or talk == "เดินหน้า":
                        print(talk)
                        tts = gTTS(text=talk,lang='th')
                        tts.save("output.mp3")
```

```
            os.system("mpg321 output.mp3")
            ser.write(b"FW\n")
            time.sleep(1)
            ser.reset_input_buffer()

        elif talk == "backward" or talk == "back" or talk == "ถอย" or talk == "ถอยหลัง":
            print(talk)
            tts = gTTS(text=talk,lang='th')
            tts.save("output.mp3")
            os.system("mpg321 output.mp3")
            ser.write(b"BW\n")
            time.sleep(1)
            ser.reset_input_buffer()

        elif talk == "turn left" or talk == "left" or talk == "ซ้าย" or talk == "เลี้ยวซ้าย":
            print(talk)
            tts = gTTS(text=talk,lang='th')
            tts.save("output.mp3")
            os.system("mpg321 output.mp3")
            ser.write(b"TL\n")
            time.sleep(1)
            ser.reset_input_buffer()

        elif talk == "turn right" or talk == "right" or talk == "ขวา" or talk == "เลี้ยวขวา":
            print(talk)
            tts = gTTS(text=talk,lang='th')
            tts.save("output.mp3")
            os.system("mpg321 output.mp3")
            ser.write(b"TR\n")
            time.sleep(1)
            ser.reset_input_buffer()

        elif talk == "สไลด์ไปทางขวา" or talk == "สไลด์ขวา" or talk == "สไลด์ไปขวา" or talk == "slide right":
            print(talk)
            tts = gTTS(text=talk,lang='th')
            tts.save("output.mp3")
            os.system("mpg321 output.mp3")
            ser.write(b"SR\n")
            ser.reset_input_buffer()

        elif talk == "สไลด์ไปทางซ้าย" or talk == "สไลด์ซ้าย" or talk == "สไลด์ไปซ้าย" or talk == "slide left":
            print(talk)
            tts = gTTS(text=talk,lang='th')
            tts.save("output.mp3")
            os.system("mpg321 output.mp3")
            ser.write(b"SL\n")
            ser.reset_input_buffer()

        elif talk == "light off" or talk == "ปิดไฟ":
            print(talk)
            tts = gTTS(text=talk,lang='th')
            tts.save("output.mp3")
            os.system("mpg321 output.mp3")
            ser.write(b"RGBoff\n")
            ser.reset_input_buffer()

        elif talk == "light on" or talk == "เปิดไฟ":
            print(talk)
            tts = gTTS(text=talk,lang='th')
            tts.save("output.mp3")
```

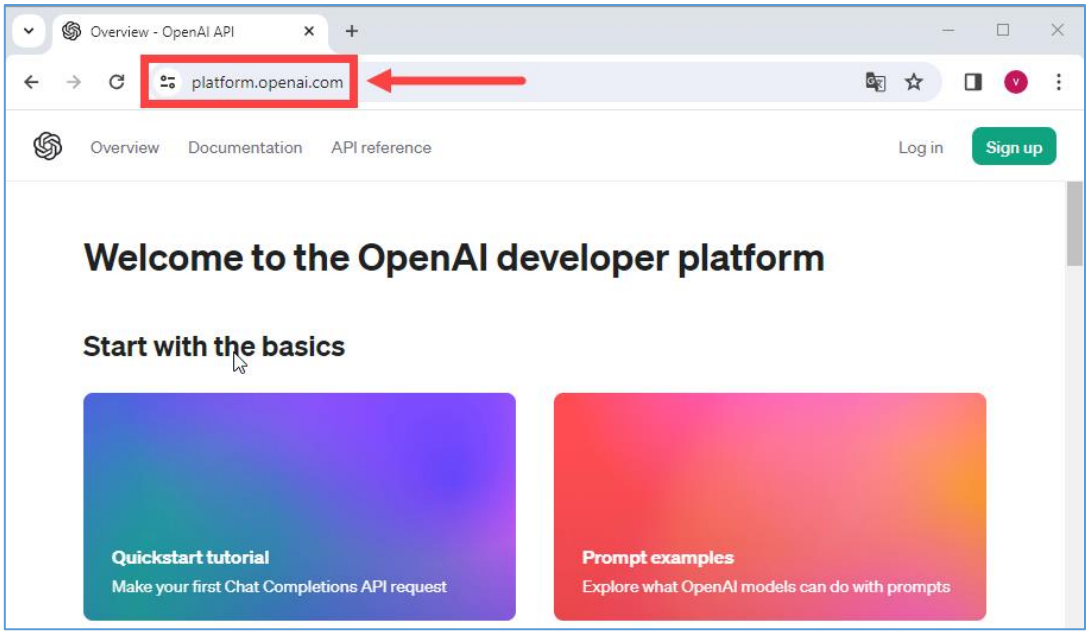
```
                    os.system("mpg321 output.mp3")
                    ser.write(b"Green\n")
                    ser.reset_input_buffer()

                elif talk == "hand up" or talk == "ยกแขน" or talk == "ยกแขนขึ้น":
                    print(talk)
                    tts = gTTS(text=talk,lang='th')
                    tts.save("output.mp3")
                    os.system("mpg321 output.mp3")
                    ser.write(b"HU\n")
                    time.sleep(1)
                    ser.reset_input_buffer()

                elif talk == "hand down" or talk == "เอาแขนลง" or talk == "แขนลง":
                    print(talk)
                    tts = gTTS(text=talk,lang='th')
                    tts.save("output.mp3")
                    os.system("mpg321 output.mp3")
                    ser.write(b"HD\n")
                    time.sleep(1)
                    ser.reset_input_buffer()

                elif talk == "exit" or talk == "ออก" or talk == "หยุด" or talk == "stop":
                    print(talk)
                    tts = gTTS(text=talk,lang='th')
                    tts.save("output.mp3")
                    os.system("mpg321 output.mp3")
                    ser.write(b"RGBoff\n")
                    time.sleep(1)
                    ser.write(b"HSTOP\n")
                    time.sleep(1)
                    ser.write(b"STOP\n")
                    time.sleep(1)
                    ser.reset_input_buffer()
                    break

            except sr.UnknownValueError:
                print("ไม่สามารถรับคำถามได้")
                #break
            except sr.RequestError as e:
                print("ข้อผิดพลาดในการเชื่อมต่อ")
                break
```

6. เขียนคำสั่ง Chatbot สอบถามข้อมูลต่างๆ จากหุ่นยนต์โดยใช้ ChatGPT ด้วยเสียงพูด

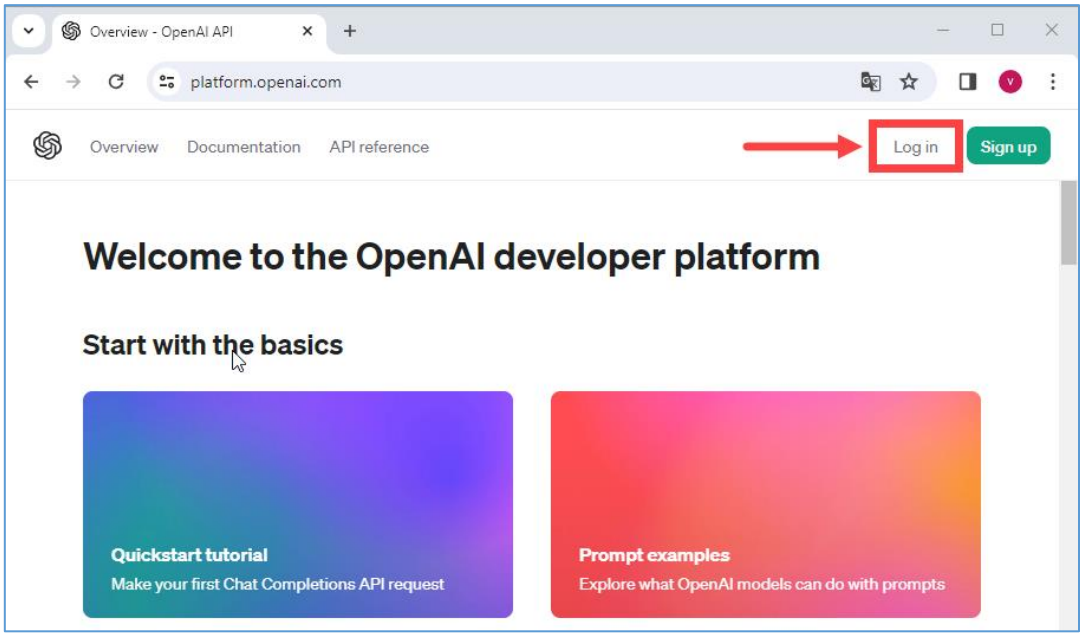
6.1 ติดตั้งไลบรารี openai โดยเปิด Terminal แล้วพิมพ์คำสั่ง pip install openai

6.2 สร้าง API Key จาก ChatGPT

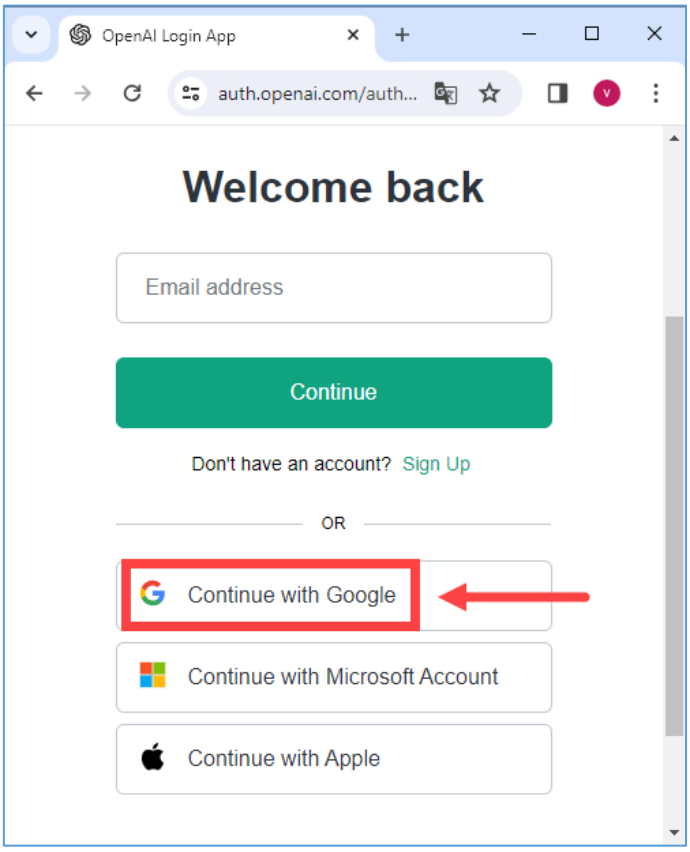
1) เข้าเว็บไซต์ https://platform.openai.com/



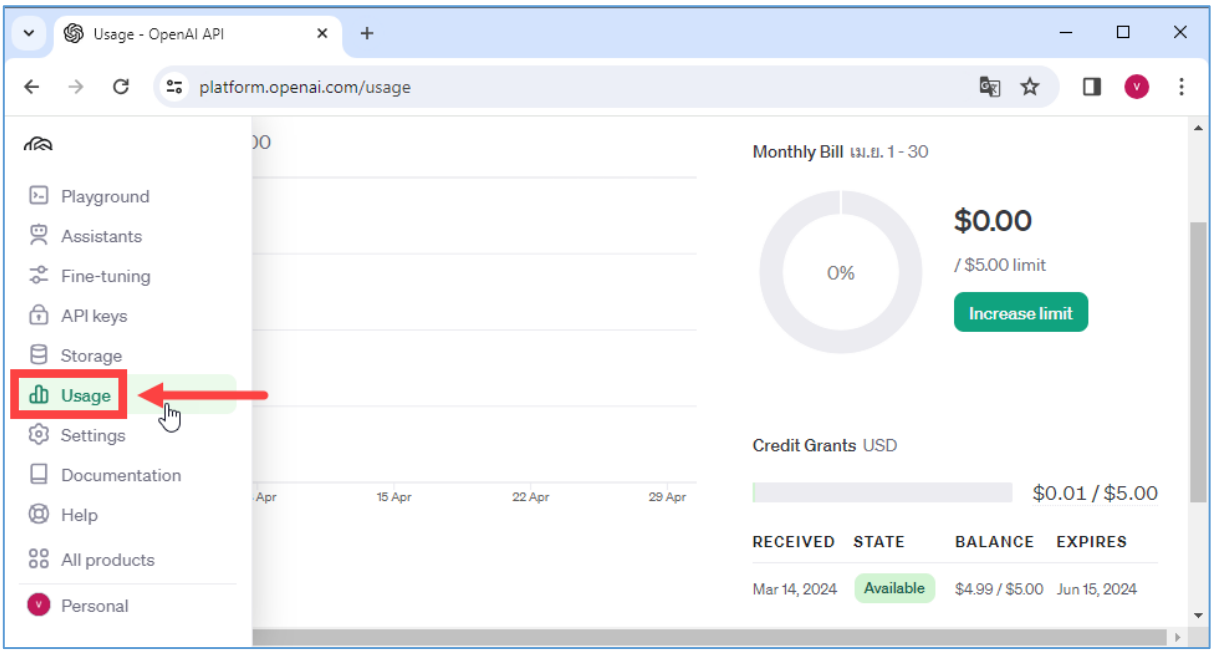
2) ล็อกอินเข้าสู่ระบบ โดยคลิกเมนู Log in



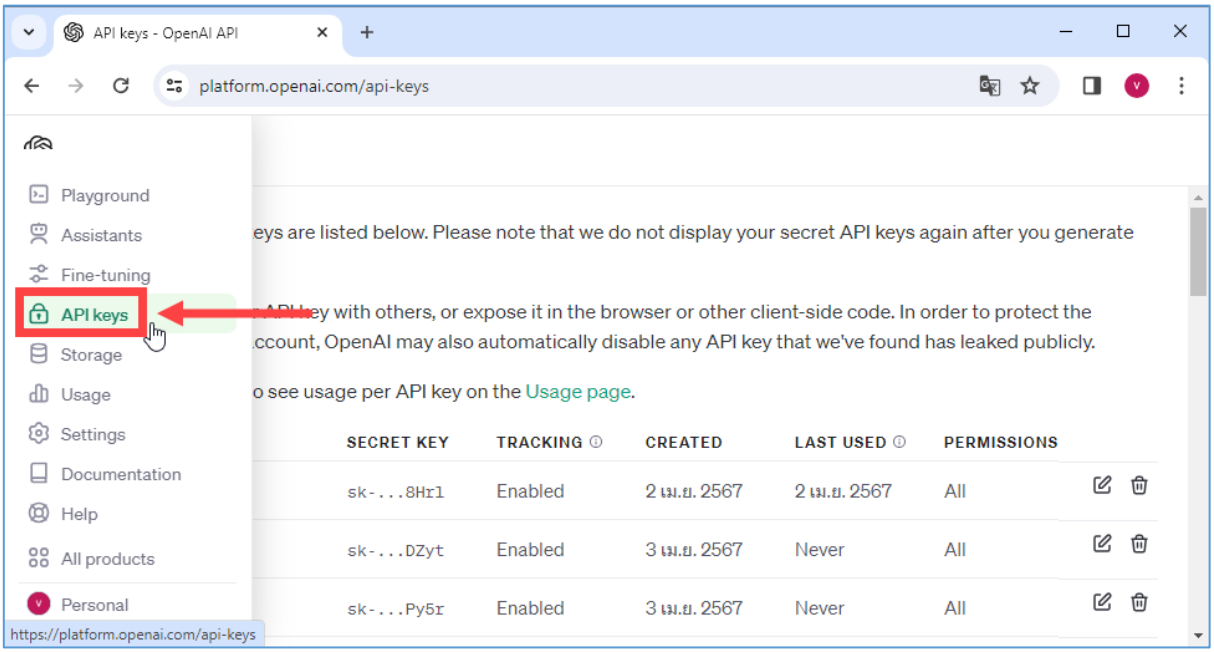
3) คลิกที่ Continue with Google



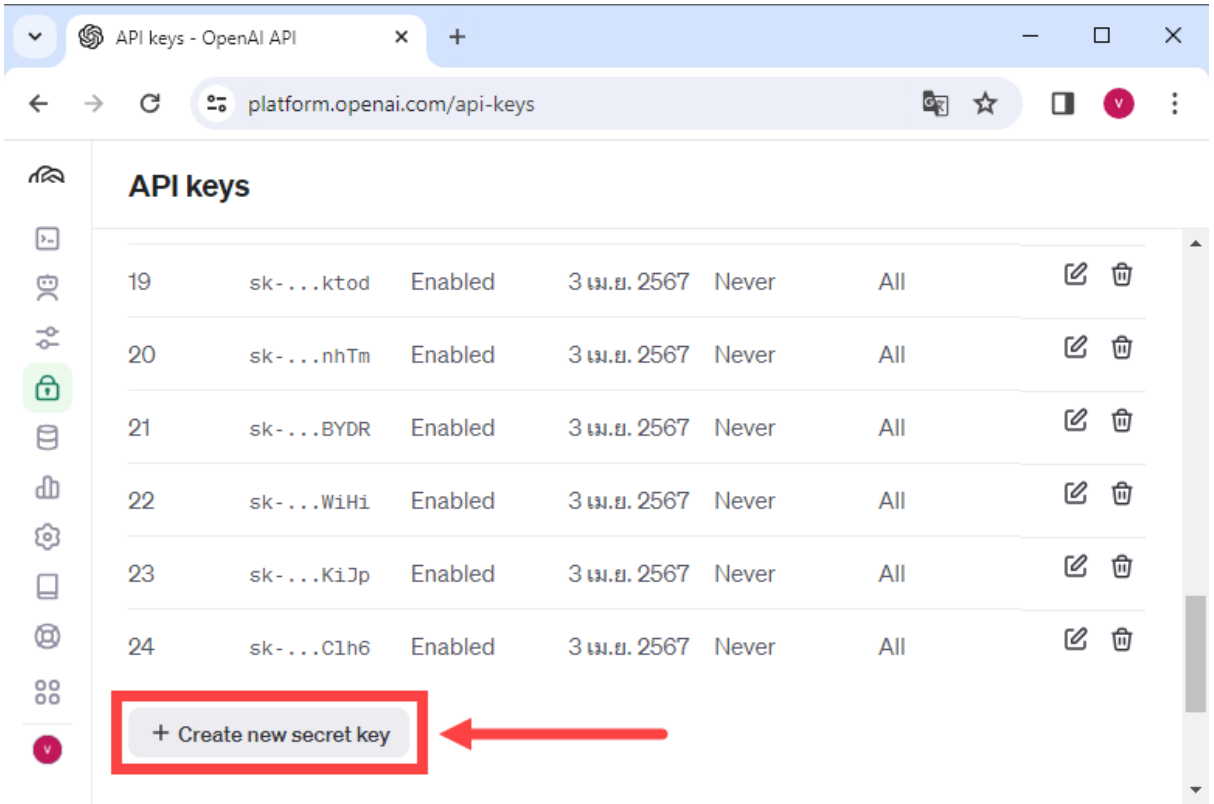
4) ตรวจสอบเครดิตคงเหลือ โดยคลิกเมนู Usage จะต้องมียอดเงินคงเหลือ และสถานะเป็น Available



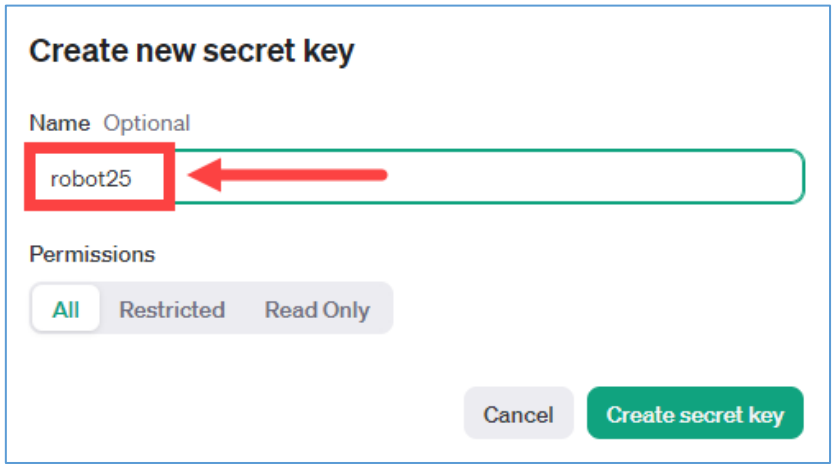
5) คลิกเมนู API keys



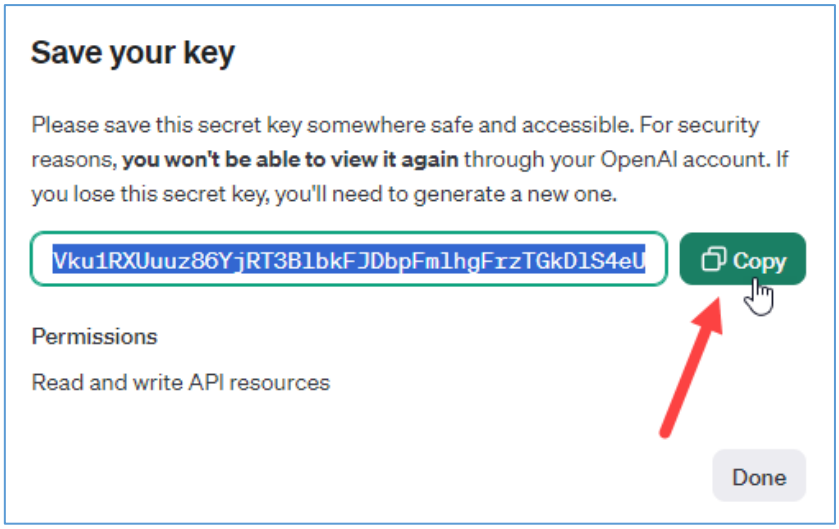
6) สร้าง API Key โดยคลิกปุ่ม + Create new secret key โดยสร้างได้สูงสุด 25 API Keys



7) ตั้งชื่อ API Key จากนั้นคลิกปุ่ม Create secret key



8) คัดลอก API Key โดยคลิกปุ่ม Copy (คัดลอก key ได้แค่ครั้งเดียว) จากนั้นคลิกปุ่ม Done



6.3 ทดสอบเขียนโปรแกรม Chatbot ที่ถาม-ตอบแบบไม่ต่อเนื่อง บันทึกไฟล์เป็น “chatbot1.py”

```python
import os
from openai import OpenAI

#Api Key ของแบงค์
client = OpenAI(api_key="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",)
model = "gpt-3.5-turbo"  # โมเดลภาษา
prompt = "สวัสดี! มีอะไรให้ฉันช่วยไหม?" # ข้อความเริ่มต้น

# ฟังก์ชัน chatbot
def chatbot():
    while True:
        print(prompt) # แสดงข้อความ prompt
        user_input = input() # รับข้อความจากผู้ใช้

        # ตรวจสอบว่าผู้ใช้ต้องการออกจากโปรแกรม
        if user_input.lower() == "quit":
            break

        # ส่งข้อความไปยัง OpenAI API
        response = client.chat.completions.create(
            messages=[
                {
                    "role": "user",
                    "content": user_input,
                }
            ],
            model=model,
        )
        print(response.choices[0].message.content) # แสดงข้อความตอบกลับจาก OpenAI

chatbot() #รัน chatbot
```

6.4 ทดสอบเขียนโปรแกรม Chatbot ที่ถาม-ตอบแบบต่อเนื่อง บันทึกไฟล์เป็น “chatbot2.py”

```python
import os
from openai import OpenAI

#Api Key ของแบงค์
client = OpenAI(api_key="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",)
model = "gpt-3.5-turbo"  # ใช้โมเดล GPT-3 ที่ใช้งานได้

# สร้างฟังก์ชัน chatbot
def chatbot():
    chat_history = []  # รักษาประวัติการสนทนา
    print("สวัสดี! มีอะไรให้ฉันช่วยไหม? (พิมพ์ 'quit' เพื่อออก)")

    while True:
        user_input = input("You: ")

        # ตรวจสอบว่าผู้ใช้ต้องการออกจากโปรแกรม
        if user_input.lower() == "quit":
            break

        # อัพเดตประวัติการสนทนาด้วยข้อความจากผู้ใช้
        chat_history.append({"role": "user", "content": user_input})

        # ส่งข้อความไปยัง OpenAI API
        response = client.chat.completions.create(
            model=model,
            messages=chat_history,
        )

         # อัพเดตประวัติการสนทนาด้วยข้อความตอบกลับจากโมเดล
         #print(response.choices[0].message.content)
         chat_response = response.choices[0].message.content
         chat_history.append({"role": "system", "content": chat_response})

         # แสดงข้อความตอบกลับจาก OpenAI
         print(f"Robot: {chat_response}")

# รัน chatbot
chatbot()
```

6.5 เขียนคำสั่ง “ถาม-ตอบด้วย Chatbot โดยใช้เสียง”

1) นำเข้าไลบรารี

```
1  import sys, os
2  from PyQt5.QtGui import *
3  from PyQt5.QtWidgets import *
4  from PyQt5.QtCore import *
5  from PyQt5.uic import *
6  import pygame
7  import time
8  import serial
9  import speech_recognition as sr
10 from gtts import gTTS
11 from openai import OpenAI
12
```

2) เขียนโค้ดคำสั่ง Chatbot ในคลาส Chatbot()

```
225 class Chatbot(QDialog): #ถาม-ตอบด้วยแชทบอต
226     def __init__(self):
227         super(Chatbot, self).__init__()
228         loadUi("chatbot.ui", self)
229         self.btnStart.clicked.connect(self.Chat)
230         self.btnHome.clicked.connect(self.Home)
231
232         # Loading the GIF
233         self.movie = QMovie("chatbot/chat2.gif")
234         self.lblRobot.setMovie(self.movie)
235         self.movie.start()
236
237     @pyqtSlot()
238     def Home(self):
239         widget.addWidget(MainWindow())
240         widget.setFixedHeight(550)
241         widget.setFixedWidth(1024)
242         widget.setCurrentIndex(widget.currentIndex()+1)
243
244     def Chat(self):
245         #Api Key
246         client = OpenAI(api_key="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",)
247         model = "gpt-3.5-turbo"  # เปลี่ยนเป็นโมเดล GPT-3 ที่ใช้งานได้
248
249         #หุ่นยนต์พูด (Text to Speech)
250         robottalk = "สอบถามข้อมูลได้เลยค่ะ"
251         tts = gTTS(text=robottalk,lang='th')
252         tts.save("output.mp3")
253         os.system("mpg321 output.mp3")
```

```
254
255         while True:
256             # รับเสียงจากผู้ใช้ (Speech to Text)
257             try:
258                 with sr.Microphone() as source:
259                     chat_history = []  # รักษาประวัติการสนทนา
260                     print("สอบถามข้อมูลได้เลยค่ะ")
261                     audio = recognizer.listen(source)
262                     question = recognizer.recognize_google(audio,language="th-TH")
263                     print(question)
264 
265                     if question == "ออก" or question == "goodbye" or question == "quit" or question == "exit":
266                         break
267 
268                     chat_history.append({"role": "user", "content": question})
269 
270                     response = client.chat.completions.create(
271                         model=model,
272                         messages=chat_history,
273                     )
274                     # แสดงข้อความตอบกลับจาก OpenAI และพูดออกมา
275                     answer = response.choices[0].message.content
276                     chat_history.append({"role": "system", "content": answer})
277                     # แสดงข้อความตอบกลับจาก OpenAI
278                     print(f"Robot: {answer}")
279                     tts = gTTS(text=answer,lang='th')
280                     tts.save("output.mp3")
281                     os.system("mpg321 output.mp3")
282 
```