```
' ================================================================================
'   {$STAMP BS2}
'   {$PBASIC 2.5}
'   TEAM 12 MECHATRONICS AND ROBOTICS PROJECT B.A.R.K
'   By Aiden Fitzpatrick, Mattam Vivekananda Swamy, Christian Hahn
'   Bluetooth Testing Website: https://compasstech.com.au/webble/index.html
' ================================================================================
' Initialization of all Variables and Constants
RFID_TX PIN 0      ' Connects to RFID R/W Module SIN
RFID_RX PIN 1      ' Connects to RFID R/W Module SOUT
#SELECT $STAMP
#CASE BS2, BS2E, BS2PE
T9600 CON 84
#ENDSELECT

Baud CON T9600
RFID_Read CON $01   ' Read data from specified address, valid locations 1 to 33 (5)
ADDR_Serial CON 32   ' Device Serial Number
ADDR_DeviceID CON 33   ' Device Identification

' Status/error return codes
ERR_OK CON $01   ' No errors
buf VAR Byte(8)   ' data buffer
idx VAR Byte      ' index

' Bluetooth Pin Activations
RxPin PIN 3
RxData VAR Bit
TxPin PIN 4
TxMsg VAR Bit
ndx VAR Nib
ctr VAR Byte
counter VAR Byte

' Emergency Stop Button
INPUT 2
Kill_Switch PIN 2

' Conversion Constants for Ultrasonic Distance units into Inches
InConstant CON 890
inDistance VAR Word
time VAR Word
' ================================================================================
'System Initialization of LCD Screen as well as Servo Motor
SEROUT 13, Baud, [22, 12] 'Initialize LCD
OUTPUT 14
FOR X = 1 TO 100 'Resets Servo Motor to default closed position
  PULSOUT 14, 500
  PAUSE 10
NEXT
' ================================================================================
' Primary operation loops, which handle the Distance-Detection Mode as well as the RFID Detection
Mode

Main_Loop: 'Distance Detection- Initially active, returns the distance from nearest object in Inches
SEROUT 13, Baud, [$11] 'Backlight LCD
SEROUT 13, Baud, [$0C] 'Clears all previous things from the LCD
PAUSE 5
DO UNTIL (RxData > 0)
  IF Kill_Switch = 1 THEN 'Repeated throughout Main_Loop, RFID_Loop, and the Door control functions,
catches the activation of the Kill_Switch at any point in time
    GOSUB terminate
  ENDIF
  SERIN RxPin, Baud, 100, Continue, [DEC RxData] 'Every "run" of the loop, the first 100ms are spent
searching for a Bluetooth Signal
  GOSUB MainBT 'If detected, jump to Bluetooth Door Activation
```

```
  Continue: 'If not detected, return to normal Ultrasonic distance detecting
  PULSOUT 15, 5
  PULSIN 15, 1, time
  inDistance = inConstant ** time 'Converts intaken values into Inches for output
  SEROUT 13, Baud, [128, DEC3 inDistance, " Inches", 13] 'Displays calculated value
  PAUSE 10
  IF indistance < 10 THEN 'If distance between the Ultrasonic and the nearest object is less than 10
Inches, then BS2 considers something to be infront of it
    GOSUB RFID_Loop 'If less than 10 inches, go to RFID Scanning
  ENDIF
LOOP

RFID_Loop: 'RFID Detection- Initially off, shifted into by Main_Loop when an object within 10 Inches
is detected
  SEROUT 13, Baud, [$0C]
  PAUSE 5
  SEROUT 13, Baud, ["Object detected, Reading Tag", 13] 'Display change of state message from
Distance Sensing to RFID Sensing
  counter = 0
  'The operation loop for the RFID reader is split between waiting on a Bluetooth Input and an RFID
being placed ahead of the antennae
  DO WHILE (counter < 75) AND (buf(0) <> ERR_OK) 'RFID reading waits for two cases- too long without
detection or correct ID Code detection
    counter = counter + 1
    IF Kill_Switch = 1 THEN
      GOSUB terminate
    ENDIF
    SERIN RxPin, Baud, 100, Read_Continue_1, [DEC RxData] 'Catch case for if something is detected,
but the bluetooth key is entered and the door is opened regardless of RFID detection
    GOSUB MainBT
    Read_Continue_1: 'If no Bluetooth Key  override, then normally scans RFID Tag
    SEROUT RFID_TX, Baud, ["!RW", RFID_Read, ADDR_Serial] ' Read tag's serial number
    SERIN  RFID_RX, Baud, 50, Read_Continue_2, [STR buf\5] ' Get status byte and data bytes
    Read_Continue_2:
  LOOP

'Controls the two fallthrough cases for the RFID_Loop reading- a correct code is detected (Success)
and either they waited too long or scanned the wrong key (Failure)
IF (buf(1) = 3) AND (buf(2) = 111) AND (buf(3) = 62) AND (buf(4) = 16) THEN 'Checks the serial key
for correctness. Success Case
  SEROUT 13, Baud, [$0C]
  PAUSE 5
  SEROUT 13, Baud, ["Correct Tag!", 148, "Welcome HOME", 13] 'Display message for a successful read
  GOSUB Door
ELSE 'Fail case for if either the detection took too long or the incorrect key was read
  RxData = 0 'Refreshes contents of relevant variable to prevent repeat reading of past entries
  buf(0) = 0
  buf(1) = 0
  SEROUT 13, Baud, ["Unwanted Guest!", 148, "Please Leave!", 13] 'Display message for failure
  PAUSE 2000
  SEROUT 13, Baud, [208, 219, 220, 222, 220, 222, 220, 222, 220, 222, 220, 222, 220, 222, 220, 222,
220, 222, 220, 222, 220, 222, 220, 222, 220, 222] 'Buzzes when entry is failed
  GOSUB Main_Loop
ENDIF

' ====================================================================
' Subroutines responsible for handling the Bluetooth Inputs, Door Opening/Closing, and Emergency
Termination

MainBT: 'Bluetooth opeations code, active during Main_Loop and Read_Tag
DO
  SEROUT 13, Baud, [$0C]
  PAUSE 5
  SEROUT 13, Baud, ["Bluetooth Key", 148, "Detected", 13] 'Print displays receipt of key
  PAUSE 100
  TxMsg = RxData
```

```
  IF TxMsg = 1 THEN 'Depending on input key from Bluetooth (0 = No Input, 1 = Open Door) jumps to
relevant function
    GOSUB Door
    ELSE
    GOTO Main_Loop
  ENDIF
LOOP

'Door Controls
Door: 'generalized controls for the opening and then closing of the door
  X VAR Byte
  RxData = 0
  FOR X = 1 TO 255 'Actuates Servo such that the Door is now Open and holds open for a few seconds
    IF Kill_Switch = 1 THEN
      GOSUB terminate
    ENDIF
    PULSOUT 14, 1000
    PAUSE 10
  NEXT
  FOR X = 1 TO 100 'Actuates Servo such that the Door is now Closed
    IF Kill_Switch = 1 THEN
      GOSUB terminate
    ENDIF
    PULSOUT 14, 500
    PAUSE 10
  NEXT
  buf(0) = 0 'Resets relevant variables
  buf(1) = 0
  GOSUB Main_Loop

'Killswitch code- when a Button input is given in any of the looped functions, the system
immediately terminates in-place
terminate:
  SEROUT 13, Baud, [$0C]
  PAUSE 5
  SEROUT 13, Baud, ["EMERGENCY STOP"]
  END 'Terminates program.
' ====================================================================
  ' -----[ End of File ]------------------------------------------------
```