Introduction to Arduino IDE and getting started with the ESP32 microcontroller

Part 5: Performing a calculation

Dr Ian Grout
Department of Electronic and Computer Engineering
Faculty of Science and Engineering
University of Limerick
Limerick, V94 T9PX
Ireland

Email: Ian.Grout@ul.ie

UNIVERSITY OF LIMERICK
OLLSCOIL LUIMNIGH

Introduction

- In this part, the following activity will be:

    - Performing a calculation within a *calculation function* using values received from the serial port and transmitting the results back to the PC. Student exercise to modify the walkthrough example developed in part 4.

- A string will be sent to the microcontroller from the PC that will give the values as float type numbers required to calculate a value where:

• Input value	Call the variable x and its type will be float
• Gain	Call the variable a and its type will be float
• Offset	Call the variable b and its type will be float
• Output value	Call the variable y and its type will be float

- To perform the calculation:

$$y = ax + b$$

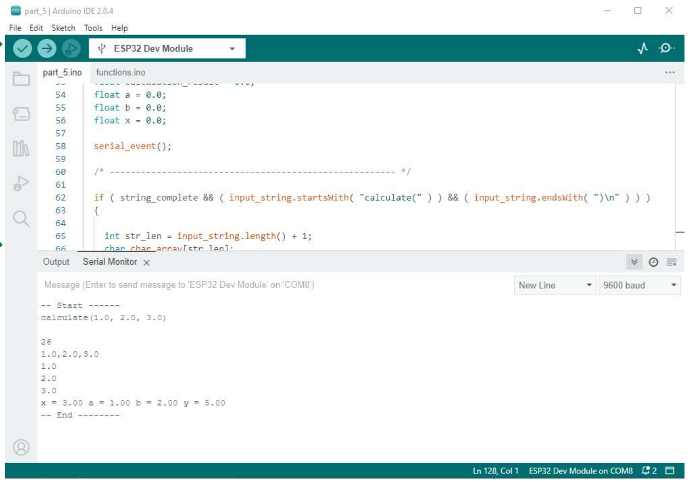
How the system works

Step 2
User creates a new, or opens an existing, Arduino sketch, selects the Arduino board and COM port to use.

Step 3
User enters the sketch code and uploads the compiled code to the microcontroller.

Step 4
User opens the Serial Monitor and sends a string to the microcontroller.

Step 6
User reads the strings received from the microcontroller.



```
float a = 0.0;
float b = 0.0;
float x = 0.0;

serial_event();

/* ------------------------------------------------------- */

if ( string_complete && ( input_string.startsWith( "calculate(" ) ) && ( input_string.endsWith( ")\n" ) ) )
{

  int str_len = input_string.length() + 1;
  char char_array[str_len];
```

Message (Enter to send message to 'ESP32 Dev Module' on 'COM8')

```
-- Start ------
calculate(1.0, 2.0, 3.0)

26
1.0,2.0,3.0
1.0
2.0
3.0
x = 3.00 a = 1.00 b = 2.00 y = 5.00
-- End --------
```

Step 1
User connects the microcontroller board to the computer

Step 5
The microcontroller receives the string from the user, performs the calculation and sends information strings back to the user.

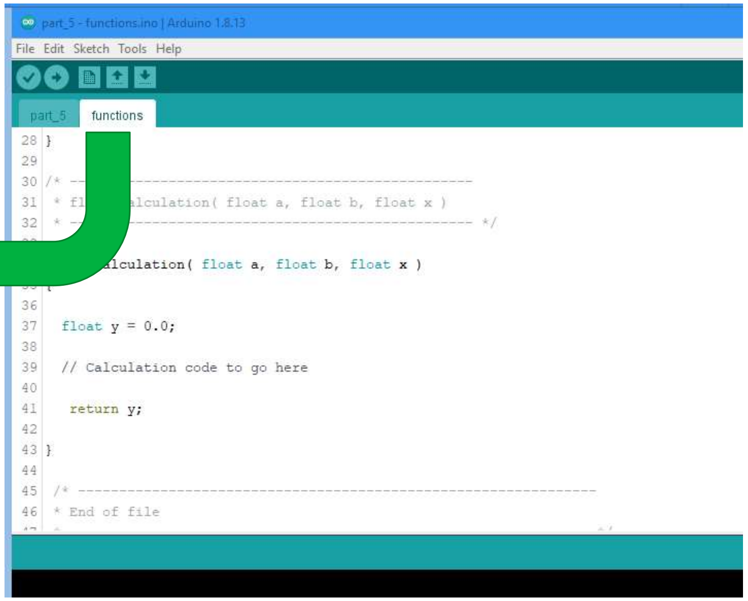
The calculation

- Use the provided Arduino sketch part_5 and complete the calculation function in order to calculate the values for y.

- Verify the code by running it on the microcontroller and send different values for a, b, and x.

```
/* ------------------------------------------------
 * float calculation( float a, float b, float x )
 * ------------------------------------------------
*/

float calculation( float a, float b, float x )
{

  float y = 0.0;

  // Calculation code to go here

   return y;

}
```

$$y = ax + b$$


part_5 - functions.ino | Arduino 1.8.13
File Edit Sketch Tools Help
part_5
functions

```
}

/* ---------------------------------------------
 * fl    alculation( float a, float b, float x )
 * ---------------------------------------------- */

      alculation( float a, float b, float x )


  float y = 0.0;

  // Calculation code to go here

   return y;

}

 /* ---------------------------------------------------------------
 * End of file
```

Extracting data from the received string (1)

- Advanced topic.
- Note the code and it's basic operation.
- Using Arduino language and C language code.
- Review, get it working, then investigate to understand.

```
serial_event();

 if ( string_complete && ( input_string.startsWith( "calculate(" ) ) && ( input_string.endsWith( ")\n" ) ) )
 {
   …
 } else if ( string_complete )
 {
   …
 } else
 {
 }
```

```
serial_event();

 if ( string_complete && ( input_string.startsWith( "calculate(" ) ) && ( input_string.endsWith( 
")\n" ) ) )
 {

   int str_len = input_string.length() + 1;
   char char_array[str_len];

   Serial.println( "-- Start ------" );
   Serial.println( input_string );
   Serial.println( str_len );
   input_string.replace( "calculate(", "" );
   input_string.replace( " ", "" );
   input_string.replace( ")\n", "" );
   Serial.println( input_string );

   input_string.toCharArray( char_array, str_len );
   token_ptr = strtok( char_array, "," );
   token_counter = 0;

   while ( token_ptr !=NULL )
   {
     Serial.println( token_ptr );
     if ( token_counter==0 )
     {
       a = atof( token_ptr );

     } else if ( token_counter==1 )
     {
       b = atof( token_ptr );

     } else if ( token_counter==2 )
     {
       x = atof( token_ptr );
     } else
     {
     }
     token_counter = token_counter + 1;
     token_ptr = strtok( NULL, "," );
   }

   calculation_result = calculation( a, b, x );

   Serial.print( "x = " );
   Serial.print( x );
   Serial.print( " a = " );
   Serial.print( a );
   Serial.print( " b = " );
   Serial.print( b );
   Serial.print( " y = " );
   Serial.println( calculation_result );
   Serial.println( "-- End --------" );

   digitalWrite( LED_BUILTIN, !digitalRead( LED_BUILTIN ) );

   input_string    = "";
   string_complete = false;

 } else if ( string_complete )
 {

   Serial.println( "Incorrect value received" );
   Serial.println( input_string );

   input_string    = "";
   string_complete = false;

 } else
 {
 }
```

Extracting data from the received string (2)

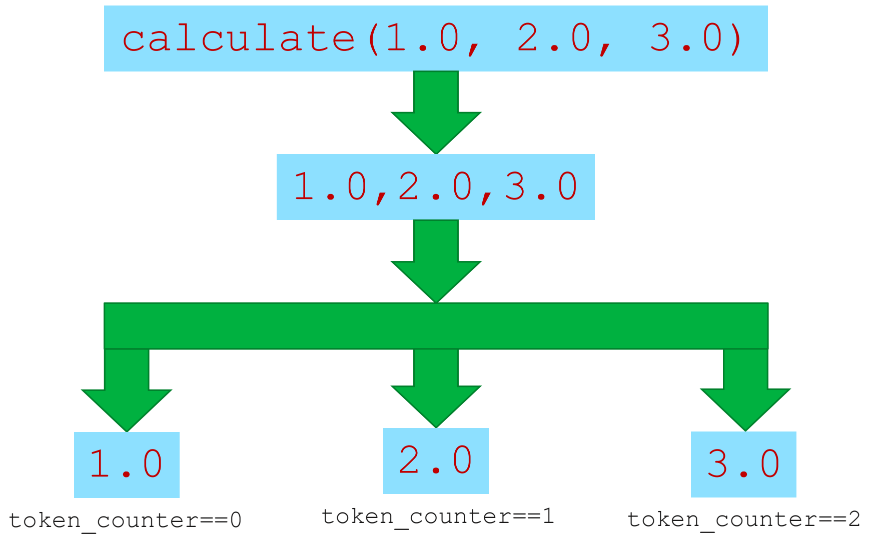
```
if ( string_complete && ( input_string.startsWith( "calculate(" ) ) && ( input_string.endsWith( ")\n" ) ) )
 {

   int str_len = input_string.length() + 1;
   char char_array[str_len];

   input_string.replace( "calculate(", "" );
   input_string.replace( " ", "" );
   input_string.replace( ")\n", "" );

   input_string.toCharArray( char_array, str_len );
   token_ptr = strtok( char_array, "," );
   token_counter = 0;

   while ( token_ptr !=NULL )
   {
     if ( token_counter==0 )
     {
       a = atof( token_ptr );

     } else if ( token_counter==1 )
     {
       b = atof( token_ptr );

     } else if ( token_counter==2 )
     {
       x = atof( token_ptr );
     } else
     {
     }
     token_counter = token_counter + 1;
     token_ptr = strtok( NULL, "," );
   }

   calculation_result = calculation( a, b, x );

   input_string    = "";
   string_complete = false;

 }
```

Extracting data from the received string (3)

```
int str_len = input_string.length() + 1;
char char_array[str_len];
```

```
input_string.replace( "calculate(", "" );
input_string.replace( " ", "" );
input_string.replace( ")\n", "" );

input_string.toCharArray( char_array, str_len );
token_ptr = strtok( char_array, "," );
token_counter = 0;
```

```
while ( token_ptr !=NULL )
{
  if ( token_counter==0 )
  {
        a = atof( token_ptr );

  } else if ( token_counter==1 )
  {
        b = atof( token_ptr );

  } else if ( token_counter==2 )
  {
        x = atof( token_ptr );
  } else
  {
  }
      token_counter = token_counter + 1;
      token_ptr = strtok( NULL, "," );
}
```

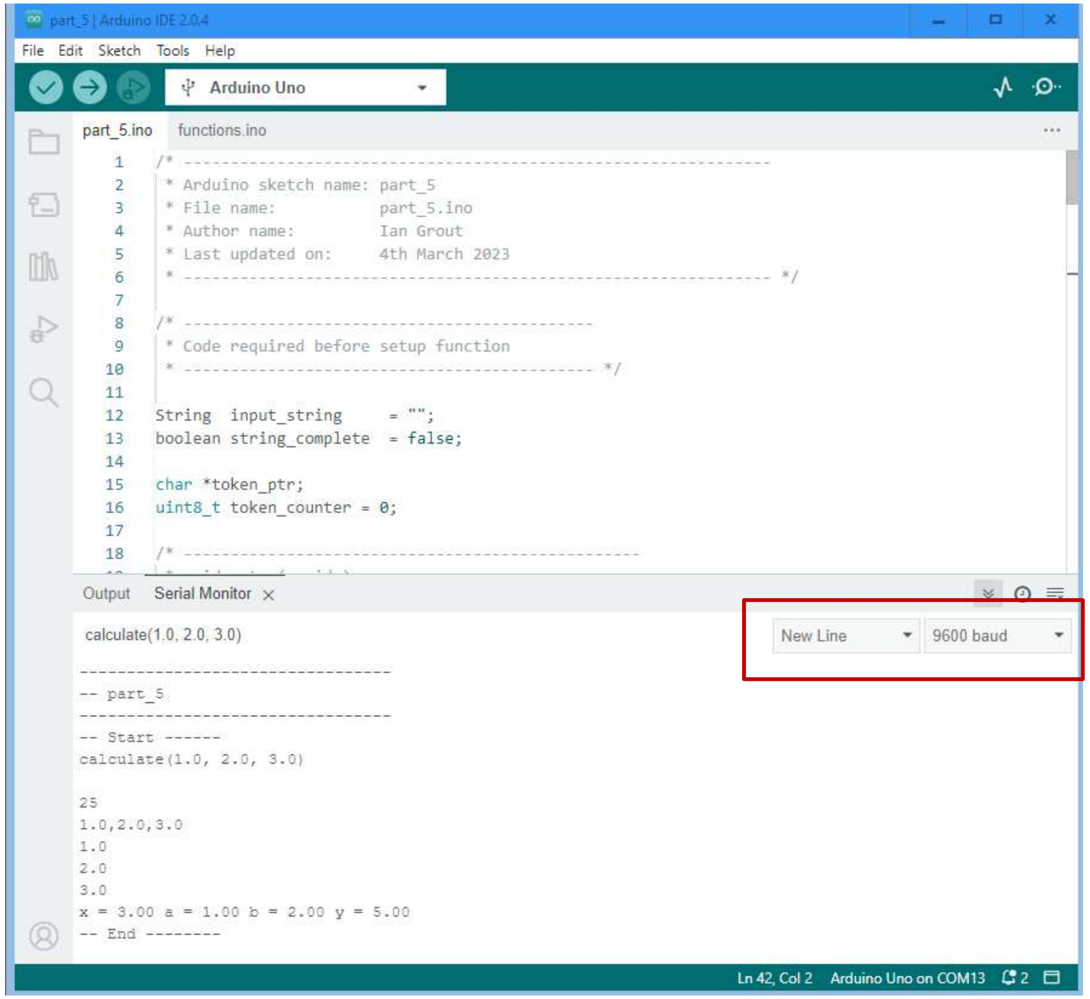


Exercise

- Obtain the part_5 Arduino sketch and complete the calculation function.
- Use the Arduino Serial Monitor to send values to the microcontroller by entering the following string:

```
calculate(1.0, 2.0, 3.0)
```

- Vary the numbers to change the values for a, b, and x.
- Watch the video part_5_video.mp4 to see the completed sketch in operation.



```
/* ---------------------------------------------------------------
 * Arduino sketch name: part_5
 * File name:           part_5.ino
 * Author name:         Ian Grout
 * Last updated on:     4th March 2023
 * --------------------------------------------------------------- */

/* ---------------------------------------------
 * Code required before setup function
 * --------------------------------------------- */

String  input_string     = "";
boolean string_complete  = false;

char *token_ptr;
uint8_t token_counter = 0;

/* ----------------------------------------------
```

```
calculate(1.0, 2.0, 3.0)
```

```
---------------------------------
-- part_5
---------------------------------
-- Start ------
calculate(1.0, 2.0, 3.0)

25
1.0,2.0,3.0
1.0
2.0
3.0
x = 3.00 a = 1.00 b = 2.00 y = 5.00
-- End --------
```

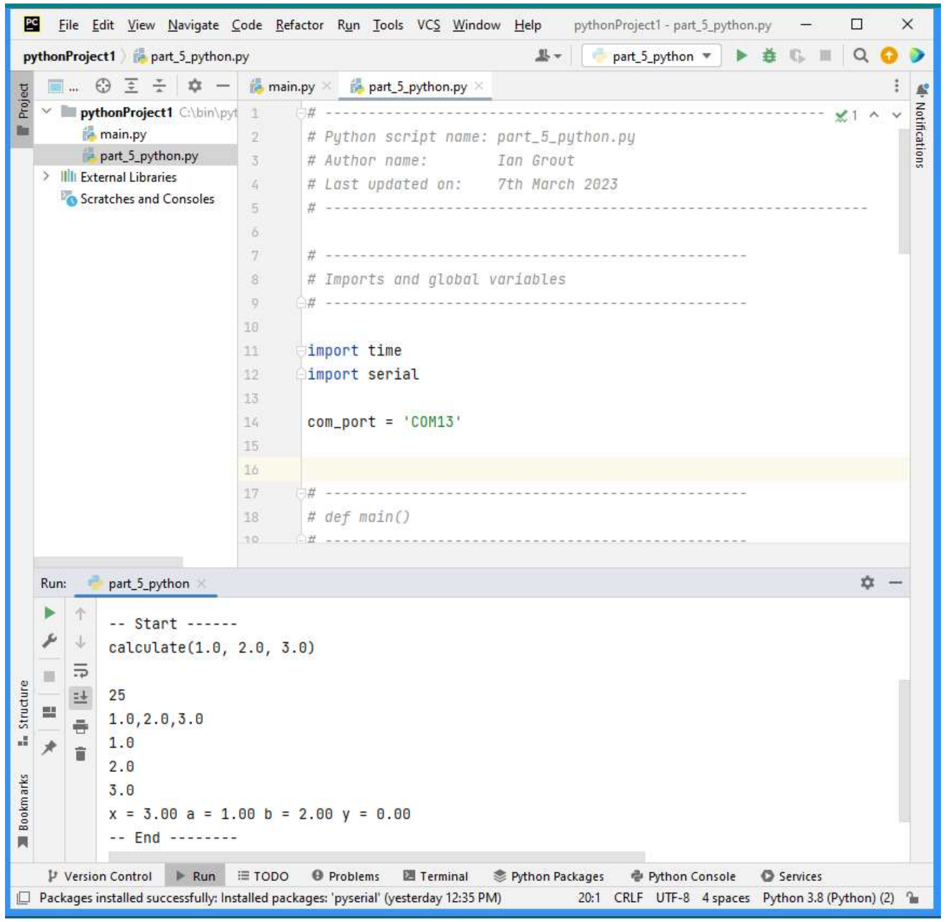
Python script to replace the Arduino IDE Serial Monitor

```
import time
import serial

com_port = 'COM8'


def main():

    ser = serial.Serial(com_port, timeout=5)
    ser.baudrate = 9600
    ser.flush()
    time.sleep(5)
    print(ser.name)

    for i in range(0, 3):
        line = ser.readline().decode('latin-1')[:-1]
        print(line)

    value_to_send = 'calculate(1.0, 2.0, 3.0)\n'

    print(value_to_send)
    ser.write(value_to_send.encode())

    time.sleep(1)

    for i in range(0, 10):
        line = ser.readline().decode('latin-1')[:-1]
        print(line)


if __name__ == '__main__':

    main()
```

- The Arduino IDE Serial Monitor is useful for initial prototyping and debugging the design code.
- For more advanced work, other software languages and tools can be used.
- For example, using Python to access the serial port as shown in the example to the left.
- This example uses pySerial to access the serial port. This is the same COM PORT as set in the Arduino IDE.
- In the code, COM8 is used on a Windows platform.

Python script to replace the Arduino IDE Serial Monitor

- Python scripts can be created and run using different software tools.
- For example, the image to the right shows the Python script developed and using PyCharm Community Edition.
- The Python script is part_5_python.py .
- Watch the video part_5_python_video to see Arduino IDE and PyCharm in use.


File Edit View Navigate Code Refactor Run Tools VCS Window Help
pythonProject1 - part_5_python.py
pythonProject1 C:\bin\pyt
main.py
part_5_python.py
External Libraries
Scratches and Consoles
Run: part_5_python
Version Control
Run
TODO
Problems
Terminal
Python Packages
Python Console
Services
20:1 CRLF UTF-8 4 spaces Python 3.8 (Python) (2)

```python
# ------------------------------------------------------------------------------
# Python script name: part_5_python.py
# Author name:        Ian Grout
# Last updated on:    7th March 2023
# ------------------------------------------------------------------------------

# ------------------------------------------------------------------------
# Imports and global variables
# ------------------------------------------------------------------------

import time
import serial

com_port = 'COM13'


# ------------------------------------------------------------------------
# def main()
# ------------------------------------------------------------------------
```

```
-- Start ------
calculate(1.0, 2.0, 3.0)

25
1.0,2.0,3.0
1.0
2.0
3.0
x = 3.00 a = 1.00 b = 2.00 y = 0.00
-- End --------
```

Packages installed successfully: Installed packages: 'pyserial' (yesterday 12:35 PM)

Any questions?

UNIVERSITY OF LIMERICK
OLLSCOIL LUIMNIGH

University of Limerick,
Limerick, V94 T9PX,
Ireland.

Ollscoil Luimnigh,
Luimneach,
V94 T9PX, Éire.

+353 (0) 61 202020

ul.ie