Challenge 1_ PWM:

The PwmOut example shows how PWM technique can be used to control the brightness of the LED. It starts by creating a PWM output object and assign pin 5 to it calling it led. After that, it runs an infinite loop which starts by incrementing the duty cycle of led by using the = operator. After each incrementation, the program waits for 200 ms. Finally, it checks whether the duty cycle of the led reached 1 in order to reset to zero.

The PWM Speaker example shows how PWM can be used to change the pitch of the sound coming out of the speaker. That is done by creating a function which controls the frequency of each tone, the volume, the interval of the signal and the rest time after the interval finishes. The volume acts as the duty cycle of the signal.

The PWM_Test_LED_NPN example is like the PwmOut example. It has only two differences. The first one is setting the frequency to 10 KHz. The second one is using the class write instead of the = operator when changing the duty cycle.

Challenge 2_ SPI:

The Temperature and Humidity example shows a simple code which keeps tracks of two sensors which are the temperature and humidity. It takes both readings. It turns LED 1 on when the temperature reading reaches 25.0 C or more.
The code uses the SPI protocol to allow the master board to communicate with the LCD. Moreover, it uses the I2C protocol to communicate with the sensors.
The code starts by clearing the screen using lcd.cls (), then getting the readings. After that, it locates where the readings should be printed on the LCD, then it prints them.

The LCD example shows how to print drawings to the LCD screen. First, it draws all shapes and stores them as data in char arrays, then it prints these drawings to the LCD screen in a specific order to give the required animation.

The SPI_Test_MOSIMISO example shows how the SPI protocol can be used to transfer data from the master device to the slave device by using the function write. Furthermore, it illustrates how to set the mode of the SPI and the frequency of the clock which synchronizing the data between the two devices. The code prints the transferred data to the pc screen which has a serial communication with.

Challenge 3_ I2C:

The temperature and Humidity example has been explained in Challenge 2.

The I2C_Test_HMC5883L example illustrates how to use the I2C protocol to set communication with a digital compass. It gets the address of the compass and stores it in the char array buffer. After that, it runs a loop to het the reading from the compass until a letter S is pressed from the pc keyboard. It reads the compass dimensions in two ways. The first one is by using the readData functions. The other on is by using getMx, getMy and getMz functions.

Challenge 4:

Here are some components and modules which can be used for building the Minimum Viable Product:

1- HMc6352 Digital Compass
2- EM406 GPS Module
3- SHT21 temperature and humidity sensor
4- RN-42 Bluetooth Module
5- ADXL345 Accelerometer
6- TMP102 temperature sensor
7- SRF08 Ultrasonic rangefinder
8- PC1602F LCD