1 Basic Programming

In this example, we intend to demonstrate the difference in using for-loop and while-loop in programming

If you haven't done so already, create a new Engduino **object** by typing in *Command Window*:

```
>> e=engduino
```

1.1 For-loop

Create a new script "forloop" by **right-click** on the "current folder" area in MATLAB. Type in the following

```
for i=0:15
    e.setLedsExact(i, e.COLOR_BLUE);
    pause(0.2);
end
```

Run the program by typing >>forloop in **Command Window,** or press **F5** in the Editor

Observe how the lights loop around exactly once. In *Command Window, type*

```
>> help pause
```

This gives you the documentation and usage example on the command *pause*

1.2 Nested for-loop

Use another for loop to make the light loop round 5 times. Enclose the code in 1.1 with

```
for outter=1:5
...
end
```

1.3 Variables

We want to make the loop go faster and faster, use a **variable** *t* to store the delay time, so that we can change it in the program.

```
t = 0.2;
for outter = 1:5
    for i=0:15
        e.setLedsExact(i,COLOR_BLUE);
        pause(t);
    end
    t = t-0.05;
end
```

1.4 Vector/array

Use a **vector/array** *c* to store the colour in each loop, use it in round to show the colour.

```
t = 0.2;
c = [e.COLOR_BLUE, e.COLOR_RED, e.COLOR_MAGENTA, e.COLOR_YELLOW, e.COLOR_GREEN];
for outter = 1:5
    for i=0:15
        e.setLedsExact(i, c(outter));
        pause(t);
    end
    t = t-0.05;
end
```

1.5 While-loop

What if we want this to loop constantly, until we want to break it with **Ctrl-C**?

Create a new script "whileloop" by **right-click** on the "current folder" area in MATLAB Type in the following:

```
i=0;
t=0.2;
while(1)
    e.setLedsExact(i, c(outter));
    pause(t);
    i = i+1;
    if i==15
        i=0;
    end
end
```

Use **Ctrl-C** in MATLAB (not the Editor) to break the infinite while loop.

Note: While loop stop only stop when the pre-defined condition is met. In the case of while(1), it never stops!

- ☐ Make it go faster and faster by introducing a variable. The speed is limited by the serial communication between MATLAB and Engduino.
- ☐ **Use the light sensor to break the while-loop** by replacing `while(1)` with `while(e.getLight)`. Add `fprintf('While-loop broken\n')` at the end of the program. You can break the while loop by covering the sensor!

1.6 Using Matrix/ 2D array

The *setLedsRgb* function takes just one input argument to change all the 16 LEDs on Engduino. It takes a 16 by 3 matrix that stored the brightness value of RGB for each LED light.

Create a new script "LEDRGB" by **right-click** on the "current folder" area in MATLAB. Use *zeros* to create a 16 x 3 matrix filled with zeros

```
b_matrix = zeros(16,3);
for inner=1:16
   b_matrix(inner, outter) = 5; %brightness
   e.setLedsRgb(b_matrix);
   pause(0.5);
```

```
end
```

Note: Matrix index in MATLAB starts from 1 (not zero!)

☐ Create an outer loop to loop through blue and green.

Let's use random number to mix our colour!

```
b_matrix = rand(16,3);                %generate uniformly distributed random numbers
b_matrix = b_matrix*8;                %normalise to interval 0, 8
b_matrix = floor(b_matrix);           %round to nearest integer
e.setLedsRgb(b_matrix);
```

1.7 Explore

In MATLAB itself, explore other Light functions

☐ e.setLedsAll(e.COLOR_BLUE)
☐ e.setLedsAllB(e.COLOR_BLUE, 2)
☐ e.setLedsOne(2, e.COLOR_BLUE)
e.setLedsOne(3, e.COLOR_RED)

2 Data Collection

The strength of integrating MATLAB and Engduino is the data collection side

2.1 Getting and plotting temperature data

In *Command Window*, type in

```
>>e.getTemperature
```

This will get you the current temperature in degree Celsius.

Create a new Script "tempExp", the follow code will update and plot the data in real time with marker 'x'

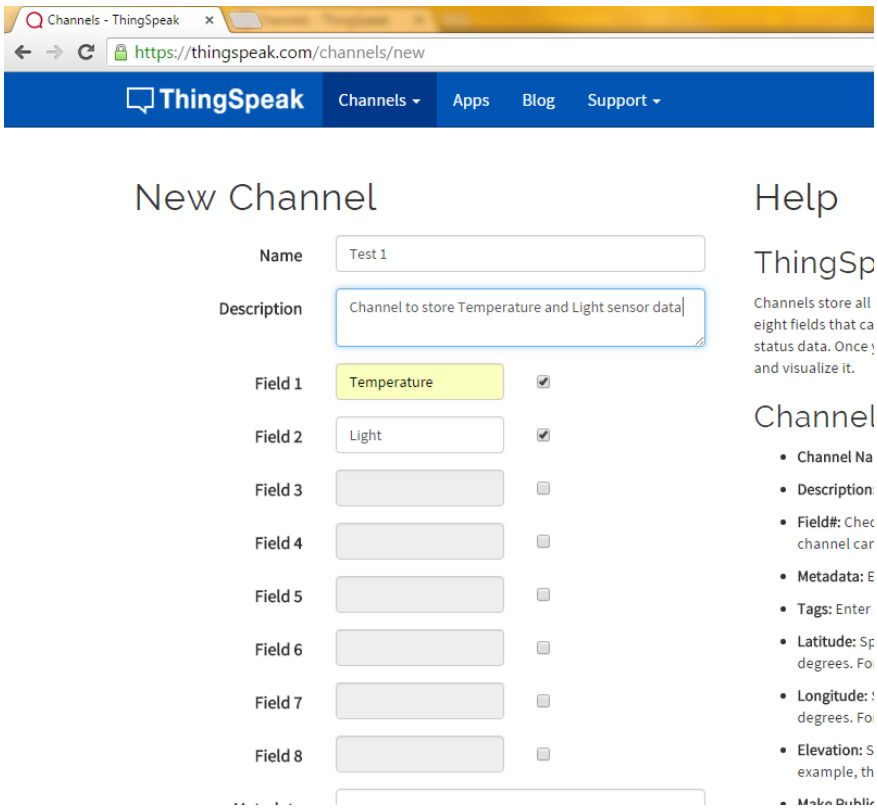
```
figure(1)
for i=1:100
    data(i) = e.getTemperature;
    plot(data(1:i), 'x-')
    refresh
    pause(1)
end
```

☐ Try plotting data from the Light sensor
☐ Try plotting data from the accelerometer

3 Data in the Cloud

Working from two, we can aim to put the temperature data onto the **Cloud,** store and visualise the data there. We use *thingspeak.com*

☐ Sign up for a ThingSpeak account at https://thingspeak.com/
☐ Create a new Channel at https://thingspeak.com/channels



☐ Save you "tempExp" script as "tempExpCloud", Add the following lines at the beginning

```
% ChId of the channel
writeChId = 59272;
% WriteKey of the channel
writeKey = 'YO5RUS9XIVVXK3JR';
```

Replace yourwriteChId with the Channel ID under "Channel Setting" tab at Thingspeak.com; writekey with Key under "API Keys" tab.

Under data(i) add the following line to get temperature data:

```
Light(i)=e.getLight;
```

Inside the for loop, add the code to upload to thingspeak.com

```
c = clock;
t = datetime(c(1),c(2),c(3),c(4),c(5),c(6));
thingSpeakWrite(writeChId, [data(i), light], ...
'Fields',[1,2],'TimeStamps',t,'Writekey',writeKey);
```

The whole code should look like this:

```
% ChId of the channel
writeChId = 59272;
% WriteKey of the channel
writeKey = 'YO5RUS9XIVVXK3JR';

figure(1)
```

```
for i=1:100
    data(i) = e.getTemperature;
    light(i)= e.getLight;
    plot([1:i], data(1:i), 'x-', [1:i], light(1:i), 'o-')
    legend('temperature', 'light')
    refresh
    pause(15)

    %upload to thingsSpeak
    c = clock;
    t = datetime(c(1),c(2),c(3),c(4),c(5),c(6));
    thingSpeakWrite(writeChId, [data(i), light(i)], ...
    'Fields',[1,2],'TimeStamps',t,'Writekey',writeKey);
end
```