

## Εργαστήριο 2

Σκοπός του συγκεκριμένου εργαστηρίου είναι η εξοικείωση με το εργαλείο mbed simulator της ARM (<https://simulator.mbed.com/>) . Το εργαλείο αυτό εξομοιώνει την αναπτυξιακή πλακέτα που θα είχαμε δια ζώσης στο εργαστήριο.

Ο προσομοιωτής σας επιτρέπει να εκτελείτε τις εφαρμογές Mbed OS 5 απευθείας στον υπολογιστή σας, ώστε να μπορείτε να ελέγχετε και να επαληθεύετε γρήγορα εφαρμογές χωρίς να τις τρέχετε σε πραγματική πλακέτα. Αυτό είναι ένα πολύτιμο εργαλείο εκμάθησης, καθώς μαθαίνετε γρήγορα πώς λειτουργεί το Mbed. Είναι επίσης πολύ χρήσιμο για την ανάπτυξη σύνθετων εφαρμογών. Στην Arm, έχουν χρησιμοποιήσει τον προσομοιωτή για εφαρμογές στο mbed-http, τη στοίβα Mbed LoRaWAN και το uTensor.

Ο προσομοιωτής δείχνει τον code editor στα αριστερά. Μπορείτε να αλλάξετε τον κωδικό σε εκείνο το πεδίο και να κάνετε κλικ στο Compile για να τον εκτελέσετε στον προσομοιωτή. Υπάρχει ένα ευρύ φάσμα διαθέσιμων demo-εφαρμογών, από περιφερειακές εφαρμογές (όπως η δημοφιλής οθόνη LCD C12832) έως εφαρμογές δικτύου. Επιλέξτε την εφαρμογή στο αναπτυσσόμενο μενού και κάντε κλικ στο Φόρτωση(Load Demo). Η εφαρμογή φορτώνεται αυτόματα.

### Πως δουλεύει

Σε αντίθεση με τα Fast Models, ο Mbed Simulator δεν προσομοιώνει πλήρως μια συσκευή Cortex-M. Αντίθετα, μεταγλωττίζει το Mbed OS 5 χρησιμοποιώντας το Emscripten. Το Emscripten είναι ένας μεταγλωττιστής LLVM σε JavaScript, ο οποίος αναλαμβάνει ένα σύνθετο έργο C ++ και παράγει κάτι που μπορεί να εκτελεστεί σε ένα περιβάλλον internet. Στη συνέχεια, όταν πάνω στην πλακέτα χρειαστεί να γίνει μια εργασία (για παράδειγμα, να αποσυνδέσουμε ένα pin ) η ενέργεια προσομοιώνεται και αντί να αλλάξουμε έναν καταχωρητή, μπορούμε να αλλάξουμε ένα γραφικό στοιχείο στην οθόνη.

Μια παρόμοια προσέγγιση ακολουθείται για τα επιμέρους κομμάτια(components). Τη στιγμή που πρέπει να πραγματοποιηθεί η πραγματική

επικοινωνία, παρεμποδίζουμε την κλήση και στέλνουμε τα δεδομένα στο πρόγραμμα περιήγησης. Για παράδειγμα, με οθόνη LCD, διακόπτουμε όταν το frame buffer της γεμίσει και οδηγούμε την εικόνα σε πίνακα από pixels. Οτιδήποτε άλλο στο πρόγραμμα οδήγησης, από την απόδοση γραμματοσειράς έως το API, παραμένει ακριβώς το ίδιο. "

Οι ασκήσεις που θα παρουσιαστούν είναι οι Blinky, Blinky with BusOut και η LCD Display οποίες είναι προ εγκατεστημένες στον online simulator. Θα περιηγηθούμε στον βασικό κώδικα των ασκήσεων να δούμε την προκαθορισμένη λειτουργία και θα επέμβουμε με δίκες μας εντολές για να δούμε την διαφορετική συμπεριφορά στην online πλακέτα.

## **Mbed OS**

Το Mbed είναι μια πλατφόρμα και λειτουργικό σύστημα , ανοικτού κώδικα για συσκευές συνδεδεμένες στο Διαδίκτυο με βάση μικροελεγκτές 32-bit ARM Cortex-M. Τέτοιες συσκευές είναι επίσης γνωστές ως συσκευές Internet of Things. Το έργο αναπτύχθηκε από κοινού από την Arm και άλλους τεχνολογικούς συνεργάτες της. Το Mbed OS περιλαμβάνει όλες τις δυνατότητες που χρειάζεστε για να αναπτύξετε ένα συνδεδεμένο προϊόν, με βάση έναν μικροελεγκτή Arm Cortex-M, όπως ασφάλεια, αποθήκευση, συνδεσιμότητα, RTOS, διαχείριση συσκευών και προγράμματα οδήγησης για αισθητήρες και συσκευές I / O.

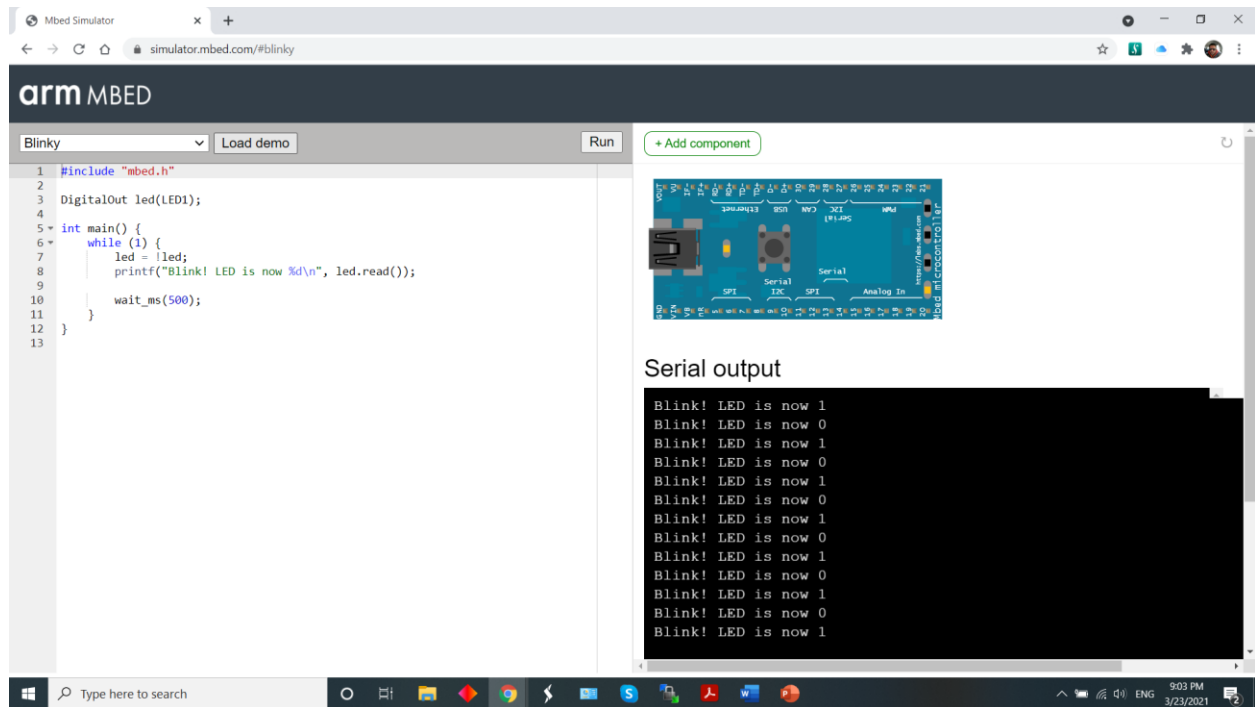
## **Τρόπος χρήσης του emulator**

Ο προσομοιωτής εμφανίζει τον code editor στα αριστερά. Μπορείτε να αλλάξετε τον κώδικα εδώ και να κάνετε κλικ στην επιλογή Μεταγλώττιση(compile) για να τον εκτελέσετε(run) στον προσομοιωτή. Υπάρχει ένα ευρύ φάσμα παραδειγμάτων, από περιφερειακά παραδείγματα (όπως η δημοφιλής οθόνη LCD C12832) έως δικτυακά παραδείγματα .Μπορείτε να χρησιμοποιήσετε την πλήρη στοίβα δικτύου Mbed απευθείας από τον προσομοιωτή. Επιλέξτε το παράδειγμα στο αναπτυσσόμενο μενού και κάντε κλικ στην επιλογή Φόρτωση(Load). Το παράδειγμα φορτώνεται αυτόματα. "

Στην συνέχεια θα ανατεθούν κάποιες ασκήσεις βάσει αυτών που θα δείξουμε για περαιτέρω παραμετροποίηση του κώδικα στον mb simulator.

Οι default κώδικες των συγκεκριμένων ασκήσεων φαίνονται παρακάτω με τις αντίστοιχες εικόνες και υπάρχουν στο <https://simulator.mbed.com/> .

# Blinky



```
#include "mbed.h"
```

```
DigitalOut led(LED1);
```

```
int main() {
```

```
    while (1) {
```

```
        led = !led;
```

```
        printf("Blink! LED is now %d\n", led.read());
```

```
        wait_ms(500);
```

```
    }
```

Η εφαρμογή Blinky είναι ένα απλό πρόγραμμα για την ενεργοποίηση και απενεργοποίηση των LED σε μια πλακέτα. Μπορεί να εκτελεστεί και σε bare metal χωρίς RTOS, αλλά στην συγκεκριμένη περίπτωση έχουμε λειτουργικό σύστημα που τρέχει από κάτω( Mbed OS 5).

## **Digitalout**

Η διεπαφή DigitalOut χρησιμοποιείται για τη διαμόρφωση και τον έλεγχο ενός ψηφιακού pin εξόδου. μπορεί να χρησιμοποιηθεί σε οποιονδήποτε pin με μπλε ετικέτα, καθώς και με τα ενσωματωμένα LED (LED1-LED4)

Η διεπαφή DigitalOut μπορεί να χρησιμοποιηθεί για να ορίσετε την κατάσταση του pin εξόδου και επίσης να διαβάσετε την τρέχουσα κατάσταση εξόδου. Ρυθμίστε το DigitalOut στο μηδέν για να το απενεργοποιήσετε ή το 1 για να το ενεργοποιήσετε.

## **wait\_ms()**

Είναι η καθυστέρηση που ορίζουμε στον κώδικα σε milliseconds. Όταν καλείτε την wait(), η CPU του mbed σας θα είναι απασχολημένη σε idle κατάσταση, μέσω του λειτουργικού συστήματος, περιμένοντας να περάσει ο απαιτούμενος χρόνος που έχετε ορίσει. Με αυτόν τον τρόπο ο προγραμματιστής του λειτουργικού συστήματος θα θέσει το τρέχον νήμα σε κατάσταση αναμονής και θα επιτρέψει σε άλλο νήμα να εκτελεστεί, ή ακόμα καλύτερα, εάν δεν υπάρχουν άλλα νήματα σε κατάσταση αναμονής, μπορεί να θέσει ολόκληρο τον μικροελεγκτή σε κατάσταση εξοικονόμησης ενέργειας.

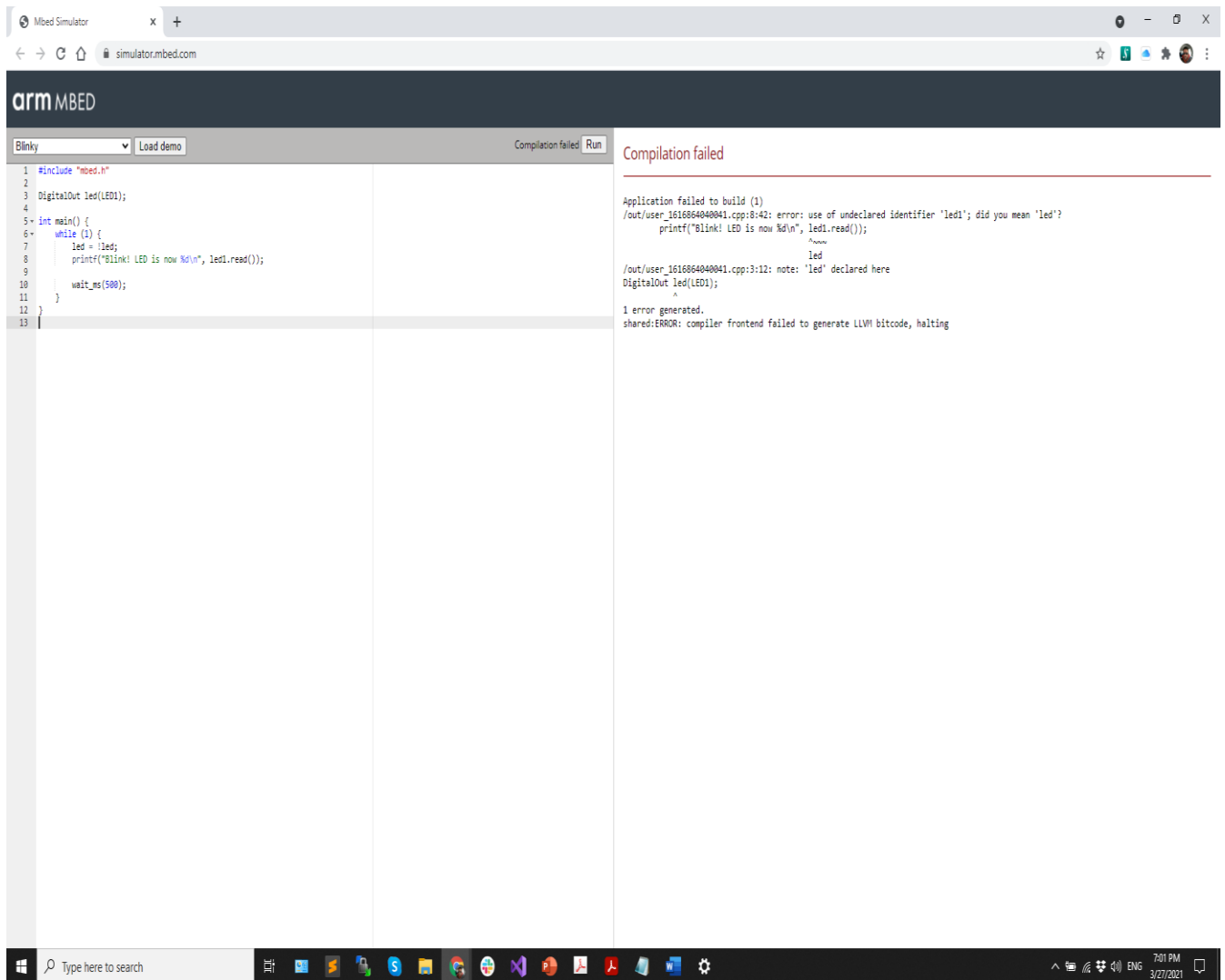
## **load demo**

Με την επιλογή αυτή φορτώνουμε το αντίστοιχο παράδειγμα που θέλουμε να τρέξουμε στην προσωρινή μνήμη.

## Run

Με την εντολή αυτή γίνεται compile και τρέχει ο κώδικας του παραδείγματος που έχουμε επιλέξει και φορτώσει.

Σε περίπτωση που ο κώδικας που εισάγουμε ή τροποποιήσουμε έχει κάποιο λάθος θα μας εμφανιστεί ένα μήνυμα όπως το παρακάτω.



The screenshot shows the Mbed Simulator web interface. The left pane contains the following C code:

```
1 #include "mbed.h"
2
3 DigitalOut led(LED1);
4
5 int main() {
6     while (1) {
7         led = !led;
8         printf("Blink! LED is now %d\n", led.read());
9     }
10    wait_ms(500);
11 }
12
13
```

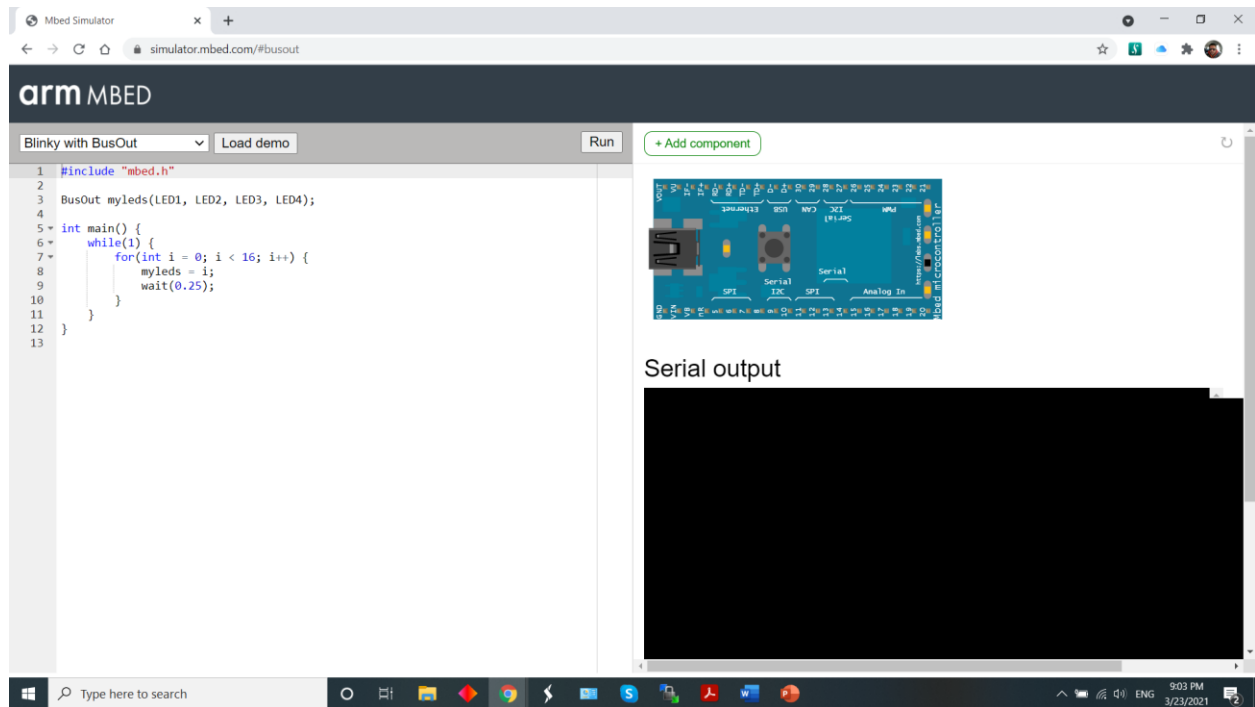
The right pane displays the compilation error message:

```
Compilation failed

Application failed to build (1)
/out/user_1616864040041.cpp:8:42: error: use of undeclared identifier 'led1'; did you mean 'led'?
    printf("Blink! LED is now %d\n", led1.read());
                                   ^~~~~
                                   led
/out/user_1616864040041.cpp:3:12: note: 'led' declared here
DigitalOut led(LED1);
           ^
1 error generated.
shared:ERROR: compiler frontend failed to generate LLVM bitcode, halting
```

The Windows taskbar at the bottom shows the time as 7:01 PM on 3/27/2021.

# Blink with BusOut



```
#include "mbed.h"
```

```
BusOut myleds(LED1, LED2, LED3, LED4);
```

```
int main() {
    while(1) {
        for(int i = 0; i < 16; i++) {
            myleds = i;
            wait(0.25);
        }
    }
}
```

Η διαφορά εδώ με την προηγούμενη εφαρμογή Blink είναι ότι έχουμε το interface(Διεπαφή) **BusOut**.

Η διεπαφή BusOut χρησιμοποιείται για τη δημιουργία ενός αριθμού pins DigitalOut που μπορούν να γραφτούν ως μία τιμή. Μπορεί να χρησιμοποιηθεί σε οποιονδήποτε ακροδέκτη με μπλε ετικέτα, καθώς και με τα ενσωματωμένα LED (LED1-LED4)

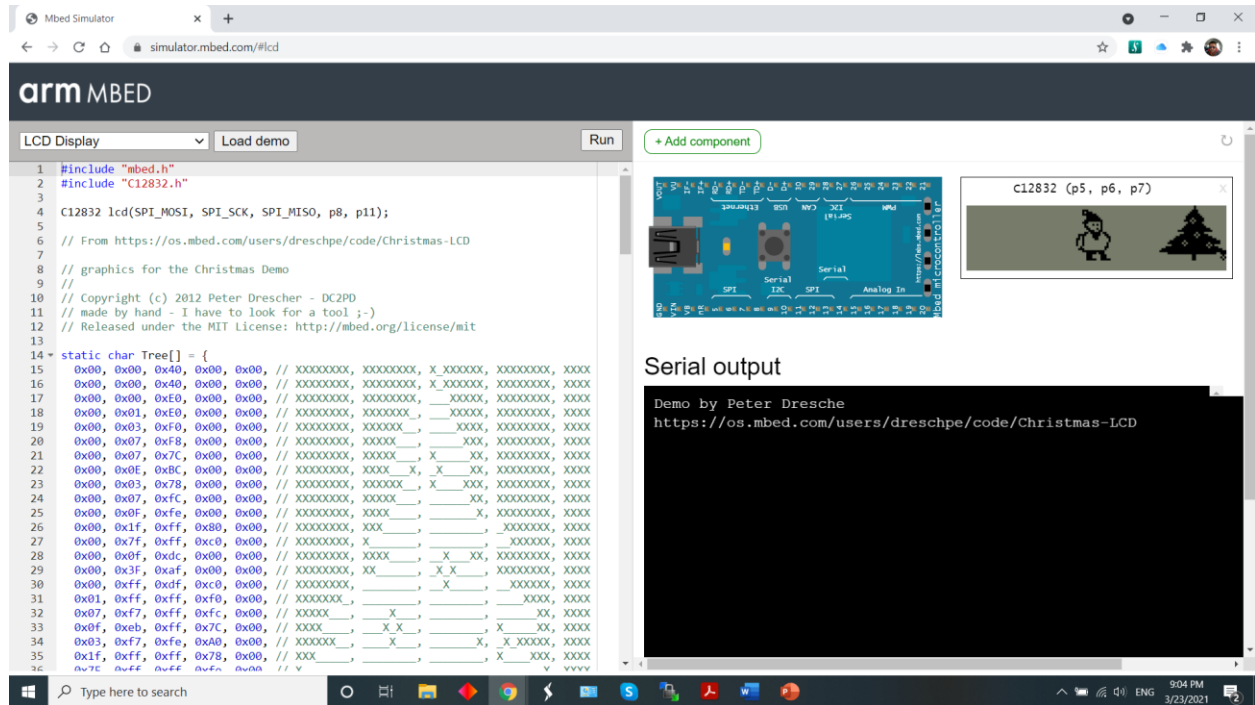
Η διεπαφή BusOut μπορεί να χρησιμοποιηθεί για να ρυθμίσετε την κατάσταση του pin εξόδου και επίσης να διαβάσετε την τρέχουσα κατάσταση εξόδου. Ρυθμίστε το BusOut στο μηδέν για να το απενεργοποιήσετε ή το 1 για να το ενεργοποιήσετε.

Στο mbed, τα τέσσερα ενσωματωμένα LED μπορούν να χρησιμοποιηθούν για τη δημιουργία διεπαφής BusOut interface και έχουν διαμορφωθεί ειδικά για να λειτουργούν χωρίς να χρειάζονται επιπλέον καλώδια ή συνδέσεις. Το BusOut είναι χρήσιμο για την εγγραφή σε πολλά pins ως ενιαία διεπαφή αντί για μεμονωμένα pins.

Μπορείτε να χρησιμοποιήσετε οποιαδήποτε από τα αριθμημένα pins Arm Mbed ως DigitalOut στο BusOut. Η διεπαφή BusOut μπορεί να χρησιμοποιηθεί για τη δημιουργία ενός τεχνητού διαύλου(Bus) από pins από οποιαδήποτε ψηφιακή είσοδο. Τα pins δεν χρειάζεται να έχουν διαδοχική διεύθυνση υλικού.



# LCD Display



```
#include "mbed.h"
```

```
#include "C12832.h"
```

```
C12832 lcd(SPI_MOSI, SPI_SCK, SPI_MISO, p8, p11);
```

```
int main() {
```

```
    lcd.cls();
```

```
    for(int i = 0; i < 150; i++) {
```

```
        lcd.locate(i, 0);
```

```
        lcd.printf("Happy");
```

```
        wait(0.1);
```

```
    }
```

```
    printf("Done!\n"); }
```





## 2) Blink with BusOut

- Χρησιμοποιήστε την printf να επαληθεύσετε τα νούμερα που εμφανίζονται μέσω των leds.

Να εμφανίζεται δηλαδή ποιο led αντιστοιχεί στο νούμερο που δείχνει και αν είναι σωστό.

- Χρησιμοποιήστε τόσο την μεταβλητή “i” όσο και την “myleds.read()”

Το i για να δείξει τον αντίστοιχο αριθμό led κάθε φορά μέσα στο loop και την myleds.read() μέσα σε printf για να τυπώσει το όνομα του αντίστοιχου Led.

- Αντικαταστήτε κάποια leds με εξωτερικά leds μέσω του “add component” button,

Για παράδειγμα αντικαταστήστε το δεύτερο εσωτερικό Led με ένα κίτρινο εξωτερικό στον ίδιο κώδικα.

- Εξετάζουμε την συμπεριφορά για counter με διαφορετικές τιμές

Αλλάζουμε την τιμή του counter i με διάφορες εναλλάξ τιμές και παρατηρούμε σε τι διαφέρει η εκτέλεση του προγράμματός μας.

## 3) LCD Display

Κατανοήστε τα παρακάτω:

- `lcd.cls();` --- Σβήνει τα πάντα από την οθόνη
- `lcd.locate(1, 5);` --- ορισμός κέρσορα , πρώτη παράμετρος η στήλης, δεύτερη παράμετρος ο αριθμός γραμμής
- `lcd.printf("Happy");` --- εκτυπώνει το μήνυμα ξεκινώντας από την τελευταία θέση του δρομέα

# ΑΣΚΗΣΕΙΣ ΓΙΑ ΠΑΡΑΔΟΣΗ

## Blinky

- Ας προσθέσουμε ένα εξωτερικό component μέσω του “Add components”
- Προσθέστε το RED LED στο p5 ακροδέκτη
- Αντικαταστήστε το “LED1” στον κώδικα με “p5”
- **Χρησιμοποιήστε τρία εξωτερικά LED ταυτόχρονα**

## Blink with BusOut

- Αντικαταστήτε όλα τα leds με εξωτερικά leds μέσω του “add component” button
- Επεκτείνετε των counter να μετράει μέχρι το 32 μέσω εξωτερικού led

## LCD Display

- Print your Name, your Family name and your A.M the one below the other
- Automatically scroll your Name for 100 columns with a for loop
- Does C12832 support scrolling from one line to the other. Experiment with this

Και τέλος μια συνδυαστική άσκηση που θα περιλαμβάνει τα ενσωματωμένα LEDES καθώς και 4 πρόσθετα εξωτερικά στοιχεία LEDs και οσοδήποτε LCD οθόνη που θα αναγράφει ένα κυλιόμενο μήνυμα της αρεσκείας.

<https://simulator.mbed.com/>