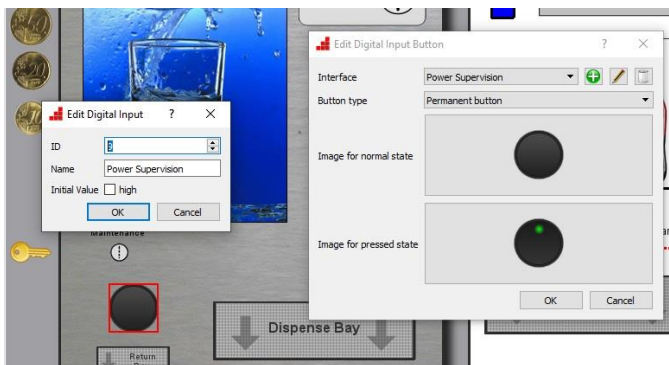


Αναφορά 2ου εργαστηρίου

Παρούσης Σταύρος Δήμος 8892 / Πουνιώτης Σταύρος 8995

Στα πλαίσια της εργασίας χρειάστηκε να υλοποιήσουμε δύο ρουτίνες c, μία που θα υλοποιεί ένα διακόπτη/κουμπί, ο οποίος θα ξεκινάει τη λειτουργία του αυτόματου πωλητή και μία όπου θα εμφανίζει τα ρέστα που χρειάζεται να επιστραφούν.

Όσον αφορά το διακόπτη/κουμπί το αρχείο το οποίο κυρίως επεξεργαστήκαμε ήταν το VendingMode.cpp. Για να γίνει αυτό αρχικά προστέθηκε στον simulator ένα digital input (κουμπί) με αρχική τιμή low και id ίσο με 9. Στη συνέχεια, μέσω της εξωτερικής συνάρτησης power() του VendingMode.cpp, “διαβάζουμε” τη τιμή του παραπάνω κουμπιού και την επιστρέφουμε σαν flag.



Με κόκκινο περίγραμμα το κουμπί που προστέθηκε

```
extern "C"{
int power() {
    int flag;
    flag = rb_sim_DigitalInput_getValue (9);
    return flag;
}
}
```

Η συνάρτηση power με τη σειρά της αξιοποιείται στη συνάρτηση enter της κλάσης VendingMode η οποία έχει τροποποιηθεί καταλλήλως, ώστε να ενεργοποιεί τη δυνατότητα να μπορούν να εισαχθούν κέρματα αφού το flag γίνει ίσο με 1, δηλαδή αφού πατηθεί το κουμπί. Επίσης με το πάτημα του κουμπιού έχουμε και την εμφάνιση του πρώτου μηνύματος στην οθόνη.

```
void VendingMode::enter() {
    int flag = power();
    while(flag==0){
        flag = power();
        if (flag == 1){
            mInner.mCashBox.setActivationState( true );
        }
    }
    mInner.mVendingState = Inner::STATE_WAITING;
    VendingScreen::drawPrompt();
}
```

Όσον αφορά τον υπολογισμό και την εμφάνιση των χρημάτων που πρέπει να επιστραφούν το αρχείο που τροποποιήσαμε ήταν το VendingScreen.cpp . Αρχικά δημιουργήσαμε τη συνάρτηση c calculate_change με ορίσματα τα λεφτά που έχει βάλει ο χρήστης και τη τιμή του προϊόντος και επιστρέφει στην οθόνη του debug τα ρέστα που πρέπει να πάρει ο χρήστης.

```
extern "C"{
u16 calculate_change(u16 price, u16 money){
    u16 change = 0;
    change = money - price;
    if (money > price){
        RB_LOG_DEBUG("Change: " << change << " cents");
    }
    return 0;
}
}
```

Τέλος σχεδιάστηκε άλλη μια συνάρτηση cpp προκειμένου να καλεί την calculate_change και να εμφανίζει τα ρέστα και στην οθόνη του αυτόματου πωλητή, η drawChangeInfo(u16 price, u16 money), η οποία με τη σειρά της καλείται από την VendingMode::Inner::onCoinsInserted.

Ο κώδικας των συγκεκριμένων συναρτήσεων επειδή καταλάμβανε μεγάλο χώρο βρίσκεται στο [Github-2ndLab](#)

Όσον αφορά τα προβλήματα που αντιμετωπίσαμε, αυτά αφορούν κυρίως την εξοικείωση με το περιβάλλον του simulator και την κατανόηση του κώδικα του Vending machine.

Για τον simulator έπρεπε πρώτα να βρούμε τα βήματα (sequence) ενεργειών που θα ακολουθούσαμε ώστε να τρέξει σωστά το simulation, στους υπολογιστές μας το εργαλείο κράσαρε συχνά, πράγμα που έκανε την διενέργεια καινούργιου simulation μετά από αλλαγές στον κώδικα κάπως ενοχλητική.

Η μεγαλύτερη πρόκληση, όπως αναφέρθηκε και στην περιγραφή της εργασίας από τους διδάσκοντες, ήταν η ανάγνωση και η κατανόηση του κώδικα του Vending machine. Ένα λάθος που κάναμε αρχικά ήταν ότι δεν χρησιμοποιήσαμε τον debugger στην διαδικασία ανάγνωσης του κώδικα και προσπαθήσαμε με τον εντοπισμό κλήσεων συναρτήσεων και κλάσεων να κατανοήσουμε την λειτουργία του προγράμματος. Μετά το εργαστήριο και την υπόδειξη του διδάσκοντα χρησιμοποιήσαμε τον debugger παράλληλα με την εκτέλεση του προγράμματος πράγμα που μας βοήθησε να κατανοήσουμε τον κώδικα και να διαμορφώσουμε την δικιά μας λύση.

Ένα από τα πράγματα που θα κρατήσουμε από την συγκεκριμένη εργασία αλλά και εργαστήριο γενικότερα είναι η τριβή που είχαμε με τα εργαλεία debugging, γενικότερα εκτιμούμε το γεγονός ότι ακόμη και από το εργαστήριο της «Αρχιτεκτονικής Υπολογιστών» του προηγούμενου εξαμήνου δίνεται έμφαση στην εξοικείωση με προγραμματιστικά εργαλεία και τεχνικές, πράγμα που βοηθάει ακόμα και τους φοιτητές που δεν ενδιαφέρονται καθαρά για το αντικείμενο.