

MyUltimateRemote

A combination of embedded and mobile. Using our CC2650 software, and our iPhone monile app, you can use any IR based operated device in your house, by tapping you phone everywhere in you house.

What is MyUltimateRemote?

The MyUltimateRemote is a combination of embedded development with mobile app creating. Our "low-level" part of the project, implemented on the CC2650 LAUNCHPAD of Texas Instruments is responsible for interfacing with the IR transmitter and Receiver - allowing us to learn how to control new devices, and controlling our existing ones. The "high level" part of this project, our iphone mobile app - "myComtroller" - let us use that IR abilities, using BLE (Bluetooth low-energy) to communicate with the CC2650.



As many devices as you can

Our mobile app allwos you to configure as many CC2650 boards as you wish, and as many Action saved on each CC2650, being as most dynamically as it can.



Mobile-friendly

The app supports all iPhone devices having ios 8.0 and above.



Easy to customise

With a change of a constant you can make your CC2650 be able to support any transimitation frequency your devices are using, and more.

Project Guidline Addressing

- ✓ Use the hardware and operating system that we used in class, or alternatives that are similar.

The entire part of the project that controls the devices, including the ability to record new devices operation signal, and the ability to operate those devices using the IR was implemented on TI CC2650 LaunchPad.

- ✓ There should be a significant software (code) development effort and challenge, not just hardware interfacing

In fact, our hardware interfacing was the minor part of the project. The main work was about understanding how to operate the IR, and designing a mechanison on the CC2650, using software development, that will act as it should - controlling real devices using the IR transmitter. Of course, in addition there was a major software development on the iPhone app, including the BLE abilities of it.

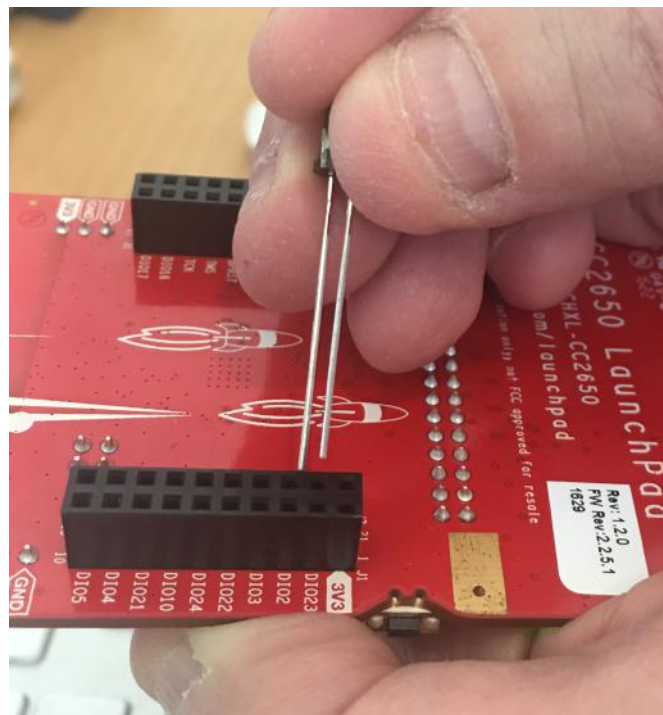
- ✓ A plausible connection to the Internet of Things

I believe that this project is truely an Internet of Things project. Using our mobile app, we can controll as many CC2650 devices as we have, making our home, workplace, campus or wherever we're at, making it a "Smart Place" - when we can controll all the devices in it using our mobile phone thanks to the MuUltimateRemote project.

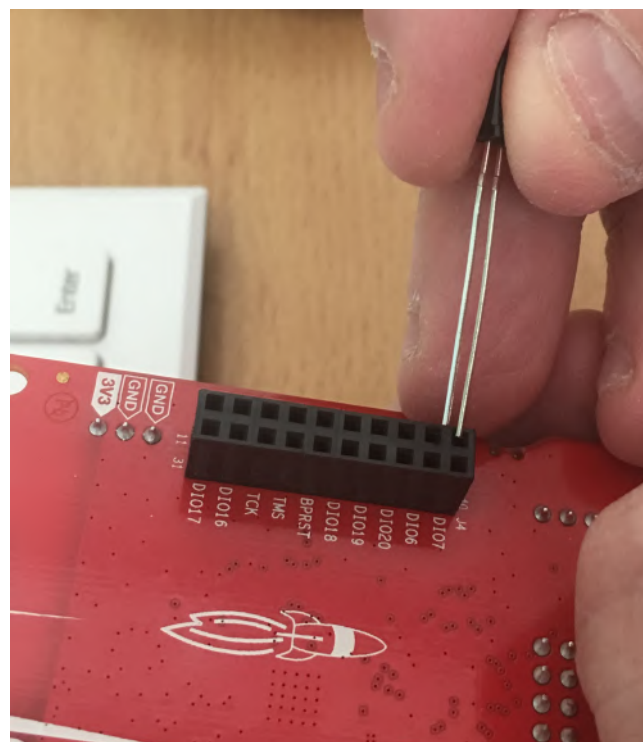
- ✓ A project that addresses a real need or has real users

חיבור הReceiver & Transmitter ללוח שלנו :

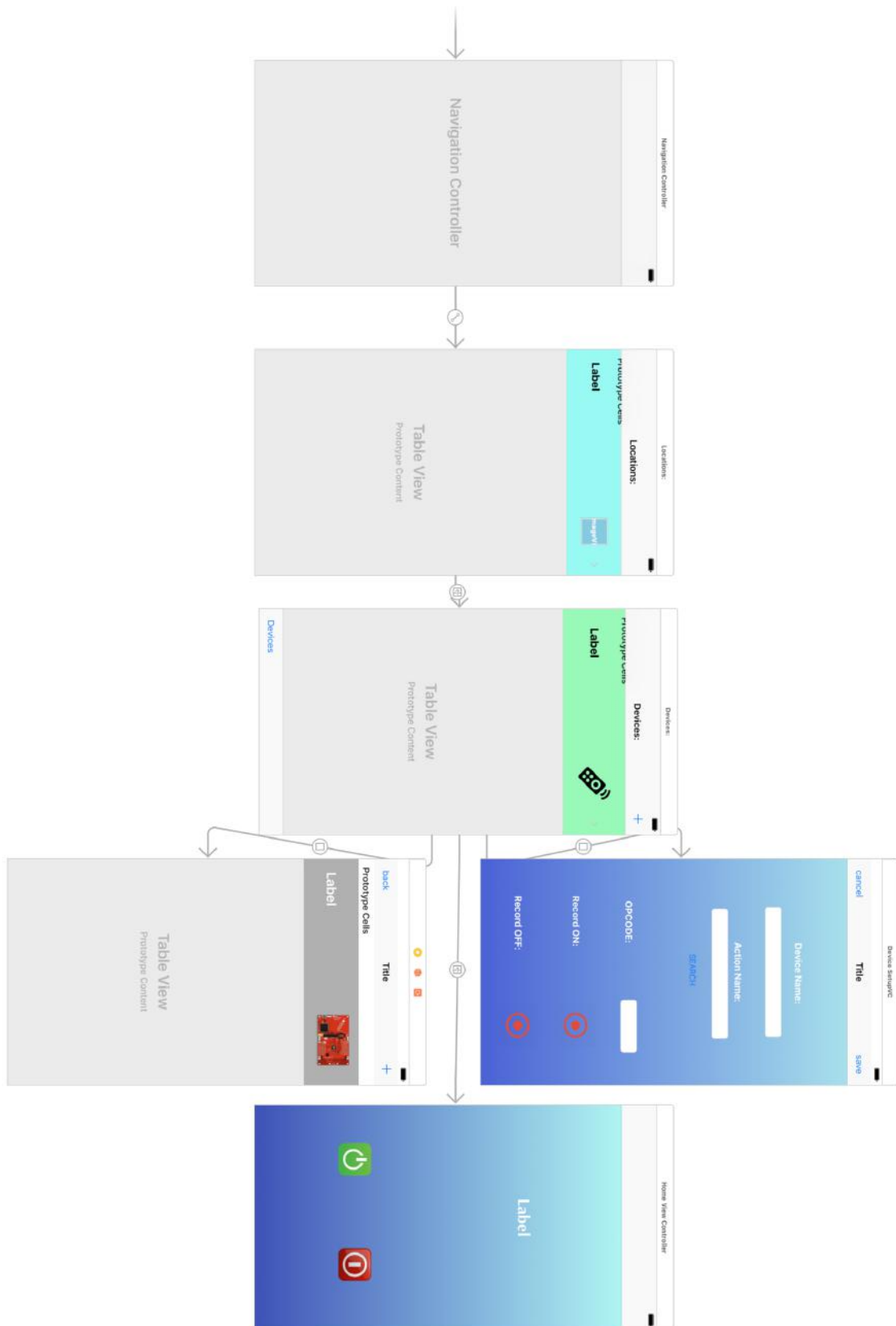
חיבור המשדר :



חיבור המקלט :



תצוגה כללית של האפליקציה:



סקירת הקוד והממשק – אפליקציית הmobilen :
(נכתבה בשפת objective-c בסביבת xcode).
תצוגת הLocations באפליקציה :



PlaceViewController.h

```
#import <UIKit/UIKit.h>

@interface PlaceViewController : UIViewController <UITableViewDelegate,
UITableViewDataSource> {

}

@end
```


PlaceViewController.m

```

#import "PlaceTVCell.h"
#import "PlaceViewController.h"
#import "DeviceViewController.h"
#import <CoreBluetooth/CoreBluetooth.h>

@interface PlaceViewController ()

@property (weak, nonatomic) IBOutlet UITableView *tableView;

@end

@implementation PlaceViewController {

    NSArray *tableData;
    NSInteger cellID;
}

- (void)viewDidLoad {
    [super viewDidLoad];
    tableData = @[@"Home - MyRoom", @"Home - LivingRoom", @"Work - Office", @"Campus - Schreiber", @"Campus - OpenSpace", @"Car"];
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
{
    return 1;
}

- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
    return [tableData count];
}

- (UITableViewCell *)tableView:(UITableView *)tableView
cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    PlaceTVCell *cell = (PlaceTVCell *)[tableView
dequeueReusableCellWithIdentifier:@"PlaceTVCell"];
    UIImage *img;
    if([indexPath section] == 0 || [indexPath section] == 1) {
        img = [UIImage imageNamed:@"home.png"];
    }
    else if([indexPath section] == 1 ) {
        img = [UIImage imageNamed:@"home.png"];
    }
    else if([indexPath section] == 2 ) {
        img = [UIImage imageNamed:@"work.png"];
    }
    else if([indexPath section] == 5 ) {
        img = [UIImage imageNamed:@"carr.png"];
    }
    else {
        img = [UIImage imageNamed:@"campus.png"];
    }
    [cell.myimage setImage:img];
    [cell.mainLabel setText:[NSString stringWithFormat:@"%s", [tableData
objectAtIndex:[indexPath section]]]];

    return cell;
}

```

MyUltimateRemote – Jhonatan Tavori

```
- (CGFloat)tableView:(UITableView *)tableView
heightForFooterInSection:(NSInteger)section{
    return 20;
}

-(UIView*)tableView:(UITableView *)tableView viewForFooterInSection:(NSInteger)section{

    UIView *footerView = [[UIView alloc] initWithFrame:CGRectMake(0, 0,
tableView.frame.size.width, 30)];
    UIColor *background = [[UIColor alloc] initWithPatternImage:[UIImage
imageName:@"10s.png"]];
    footerView.backgroundColor = background;
    return footerView;
}

- (void) tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath
*)indexPath
{
    cellID = [indexPath section];
    [self performSegueWithIdentifier:@"ChooseLocation" sender:nil];
}

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender {

    if([segue.identifier isEqualToString:@"ChooseLocation"])
    {
        DeviceViewController *devViewController = [segue destinationViewController];
        devViewController.placeid = [tableData objectAtIndex:cellID];
    }
}

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
}

@end
```

PlaceTVCell.h

```
#import <UIKit/UIKit.h>

@interface PlaceTVCell : UITableViewCell {

}

@property (weak, nonatomic) IBOutlet UILabel *mainLabel;
@property (weak, nonatomic) IBOutlet UIImageView *myimage;

@end
```

PlaceTVCell.m

```
#import "PlaceTVCell.h"

@interface PlaceTVCell ()

@end

@implementation PlaceTVCell {}

@end
```

תצוגת המכשירים (בתוך מיקום ספציפי):



DeviceViewController.h

```
#import <UIKit/UIKit.h>

@interface DeviceViewController : UIViewController <UITableViewDelegate,
UITableViewDataSource> {
}
}
```



```
@property (strong) NSString *placeid;
@property (nonatomic, strong) UITableView *tableview;
@property (nonatomic, strong) UILabel *mmmlab;
```

```
@end
```

DeviceViewController.m

```
#import "DeviceViewController.h"
#import "HomeController.h"
#import "DeviceSetupVC.h"
#import "DeviceTVCell.h"
#import "ShowDeviceListVC.h"

@interface DeviceViewController ()

@end

@implementation DeviceViewController {

    NSArray *tableData, *actData;
    NSMutableArray *actionsNames;
    NSMutableArray *devicesNames;
    NSInteger cellID;
}

- (void)viewDidLoad {
    [super viewDidLoad];
}

-(void)viewDidAppear:(BOOL)animated{

    actionsNames = [[NSMutableArray alloc] init];
    devicesNames = [[NSMutableArray alloc] init];
    tableData = [NSKeyedUnarchiver unarchiveObjectWithData:[NSUserDefaults
standardUserDefaults] objectForKey:_placeid]] ;
    NSString *name;
    int i, j;

    for (i=0; i < tableData.count; i++) {

        name = [tableData objectAtIndex:i];
        actData = [NSKeyedUnarchiver unarchiveObjectWithData:[NSUserDefaults
standardUserDefaults] objectForKey:name]] ;
        for (j=0; j < actData.count; j++) {

            [actionsNames addObject:[actData objectAtIndex:j]];
            [devicesNames addObject:name];

        }
    }

    [_tableview reloadData];
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
{
    return 1;
}

- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
```

MyUltimateRemote – Jhonatan Tavori

```
{
    return [actionsNames count];
}

- (UITableViewCell *)tableView:(UITableView *)tableView
cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    DeviceTVCell *cell = (DeviceTVCell *)[tableView
dequeueReusableCellWithIdentifier:@"DeviceTVCell"];
    [cell.mainLabel setText:[NSString stringWithFormat:@"%@" - %@", [devicesNames
objectAtIndex:[indexPath section]], [actionsNames objectAtIndex:[indexPath section]]]];
    return cell;
}

- (CGFloat)tableView:(UITableView *)tableView
heightForFooterInSection:(NSInteger)section{
    return 20;
}

-(UIView*)tableView:(UITableView *)tableView viewForFooterInSection:(NSInteger)section{

    UIView *footerView = [[UIView alloc] initWithFrame:CGRectMake(0, 0,
tableView.frame.size.width, 30)];
    UIColor *background = [[UIColor alloc] initWithPatternImage:[UIImage
imageName:@"10s.png"]];
    footerView.backgroundColor = background;
    return footerView;
}

- (void) tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath
*)indexPath
{
    cellID = [indexPath section];
    [self performSegueWithIdentifier:@"ChooseDevice" sender:nil];
}

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender {
    if([segue.identifier isEqualToString:@"ChooseDevice"])
    {
        HomeViewController *devViewController = [segue destinationViewController];
        devViewController.deviceid = [devicesNames objectAtIndex:cellID];
        devViewController.placeid = _placeid;
        devViewController.actid = [actionsNames objectAtIndex:cellID];
    }
    if([segue.identifier isEqualToString:@"showDevicesSeg"])
    {
        ShowDeviceListVC *devViewController = [segue destinationViewController];
        devViewController.placeid = _placeid;
    }
}

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
}

@end
```

DeviceTVCell.h

```
#import <UIKit/UIKit.h>

@interface DeviceTVCell : UITableViewCell {
}

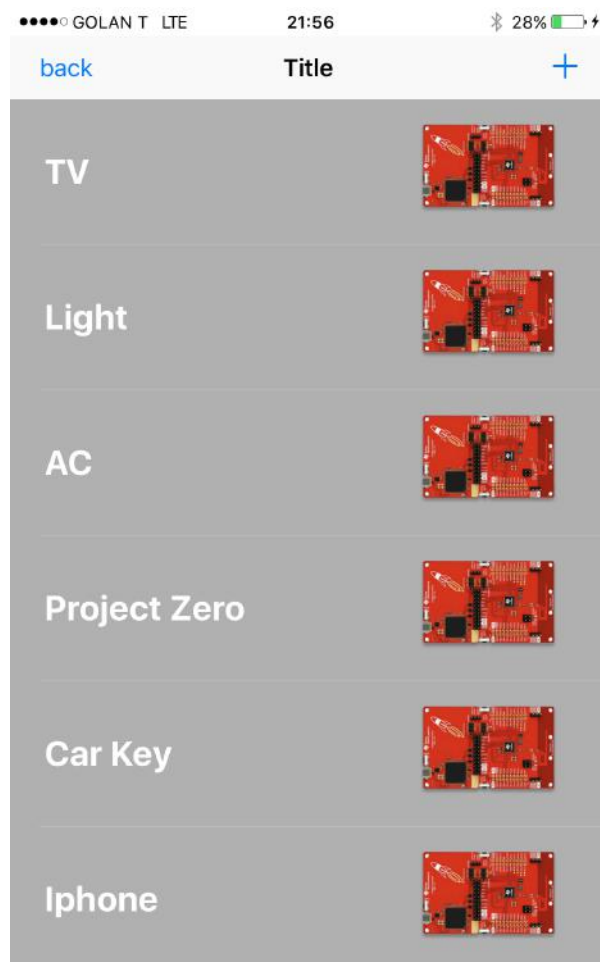
}
```

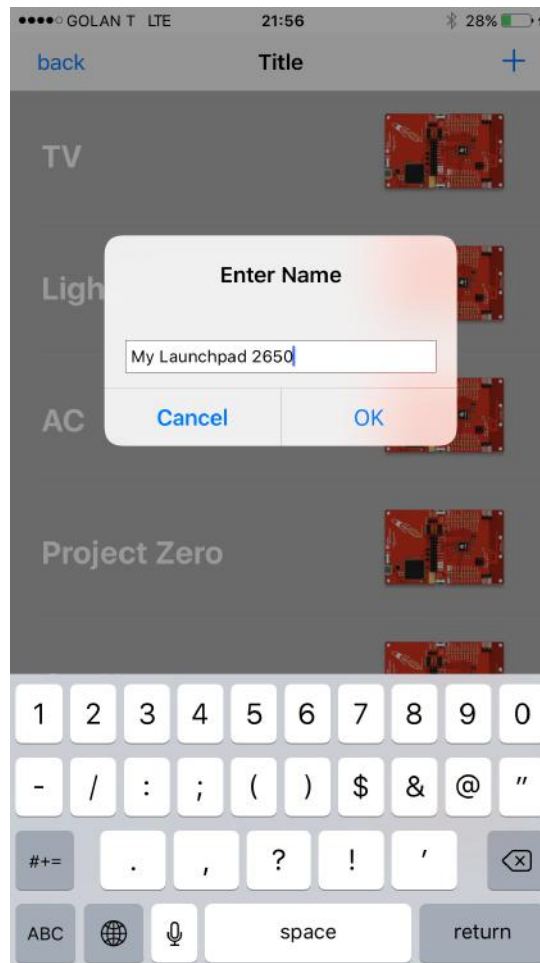
```
@property (weak, nonatomic) IBOutlet UILabel *mainLabel;  
@end
```

DeviceTVCell.m

```
#import "DeviceTVCell.h"  
  
@interface DeviceTVCell ()  
@end  
  
@implementation DeviceTVCell {}  
  
@end
```

תצוגת המכשירים (בלבד), ואפשרות הוספתם (במיקום נתון):





ShowDeviceListVC.h

```
#import <UIKit/UIKit.h>

@interface ShowDeviceListVC : UIViewController <UITableViewDelegate,
UITableViewDataSource> {
}

@property (strong) NSString *placeid;
@property (weak, nonatomic) IBOutlet UITableView *tableview;
@property (nonatomic, strong) UITableView *tableview2;

- (IBAction)cancel:(id)sender;
- (IBAction)add:(id)sender;

@end
```

ShowDeviceListVC.m

```

#import "ShowDeviceListVC.h"
#import "HomeViewController.h"
#import "DeviceSetupVC.h"
#import "DeviceTVCell.h"

@interface ShowDeviceListVC ()

@end

@implementation ShowDeviceListVC {

    NSArray *tableData;
    NSInteger cellID;
    NSMutableArray *devicesNames;

}

- (void)viewDidLoad {

    [super viewDidLoad];

    devicesNames = [[NSMutableArray alloc] init];
    tableData = [NSKeyedUnarchiver unarchiveObjectWithData:[NSUserDefaults
standardUserDefaults] objectForKey:_placeid]] ;
    NSString *name;
    int i;
    for (i=0; i < tableData.count; i++) {
        name = [tableData objectAtIndex:i];
        [devicesNames addObject:name];
    }

}

-(void)viewDidAppear:(BOOL)animated{

    devicesNames = [[NSMutableArray alloc] init];
    tableData = [NSKeyedUnarchiver unarchiveObjectWithData:[NSUserDefaults
standardUserDefaults] objectForKey:_placeid]] ;
    NSString *name;
    int i;
    for (i=0; i < tableData.count; i++) {
        name = [tableData objectAtIndex:i];
        [devicesNames addObject:name];
    }

    [_tableView reloadData];

}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
{
    return [devicesNames count];
}

- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
    return 1;
}

- (UITableViewCell *)tableView:(UITableView *)tableView
cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    DeviceTVCell *cell = (DeviceTVCell *)[tableView
dequeueReusableCellWithIdentifier:@"DeviceTVCell2"];
}

```

MyUltimateRemote – Jhonatan Tavori

```
[cell.mainLabel setText:[NSString stringWithFormat:@"%@", [devicesNames
objectAtIndex:[indexPath row]]]];
    return cell;
}

-(IBAction)cancel:(id)sender {

    [self dismissViewControllerAnimated:YES completion:nil];
}

-(IBAction)add:(id)sender {

    UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@"Enter Name"
                                                    message:@" "
                                                    delegate:self
                                                    cancelButtonTitle:@"Cancel"
                                                    otherButtonTitles:@"OK", nil];
    alert.alertViewStyle = UIAlertViewStylePlainTextInput;
    [alert show];
}

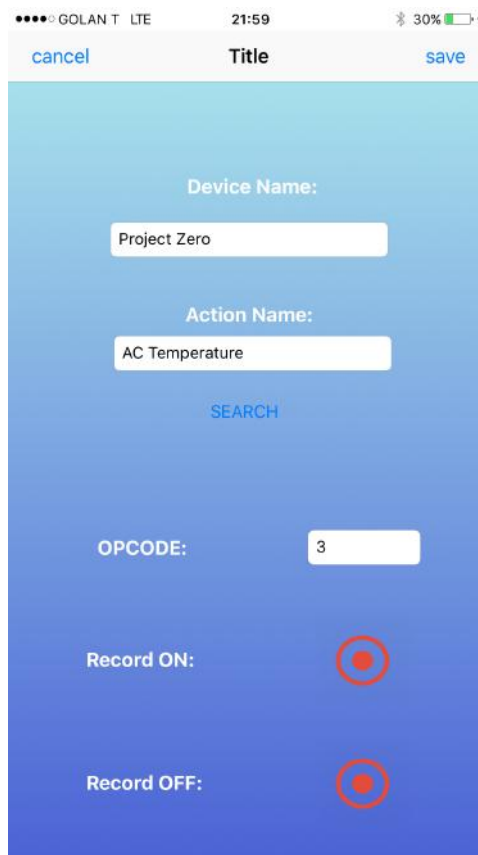
- (void)alertView:(UIAlertView *)alertView clickedButtonAtIndex:(NSInteger)buttonIndex {
    if (buttonIndex == 1) {
        NSString *name = [alertView textFieldAtIndex:0].text;
        NSArray *devarr1 = [NSKeyedUnarchiver unarchiveObjectWithData:[NSUserDefaults
standardUserDefaults] objectForKey:_placeid]] ;
        NSMutableArray *devarr2 = [NSMutableArray arrayWithArray:devarr1];
        [devarr2 addObject:name];
        devarr1 = [devarr2 copy];
        [[NSUserDefaults standardUserDefaults] setObject: [NSKeyedArchiver
archivedDataWithRootObject:devarr1] forKey:_placeid];
        NSMutableArray *ddd = [NSMutableArray alloc] init];
        [[NSUserDefaults standardUserDefaults] setObject: [NSKeyedArchiver
archivedDataWithRootObject:ddd] forKey:name];

        [devicesNames addObject:name];
        [_tableView reloadData];
    }
}

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
}

@end
```


הוספת מכשיר ופעולה, כולל הקלטה :



: DeviceSetupVC.h

```
#import <UIKit/UIKit.h>

@interface DeviceSetupVC : UIViewController {
}

@property (weak, nonatomic) IBOutlet UITextField *mybigTF;
@property (weak, nonatomic) IBOutlet UITextField *myTF;
@property (weak, nonatomic) IBOutlet UITextField *myCode;
@property (weak, nonatomic) IBOutlet UILabel *myLab;
@property (weak, nonatomic) IBOutlet UILabel *searchLab;
@property (weak, nonatomic) IBOutlet UIButton *searchBut;

@property (strong) NSString *placeId;

-(IBAction)cancel:(id)sender;
-(IBAction)save:(id)sender;

-(IBAction)search:(id)sender;

@end
```

: DeviceSetupVC.m

```

#import "DeviceSetupVC.h"
#import <CoreBluetooth/CoreBluetooth.h>
#import "Constants.h"
#import <CoreData/CoreData.h>

#define TIMER_SCAN_INTERVAL 2.0

@interface DeviceSetupVC () <CBCentralManagerDelegate, CBPeripheralDelegate>

// view
@property (weak, nonatomic) IBOutlet UIView *controlContainerView;
@property (weak, nonatomic) IBOutlet UIView *tableView;

@property (weak, nonatomic) IBOutlet UILabel *myLabel;
@property (weak, nonatomic) IBOutlet UIButton *three;
@property (weak, nonatomic) IBOutlet UIButton *five;

@property (retain) NSMutableArray *devArrr;

// bluetooth
@property (nonatomic, strong) CBCentralManager *centralManager;
@property (nonatomic, strong) CBPeripheral *cc2650l;
@property (nonatomic, strong) CBCharacteristic *redLed;
@property (nonatomic, strong) CBCharacteristic *greenLed;
@property (nonatomic, assign) BOOL keepScanning;
@property (nonatomic, assign) BOOL connected;
-(IBAction)threep:(id)sender;
-(IBAction)fivep:(id)sender;
-(IBAction)switchview:(id)sender;

@end

@implementation DeviceSetupVC {

    NSString *devName;
}

- (void)viewDidLoad {

    [super viewDidLoad];

    devName = @"NONE";
    self.searchLab.text = [NSString stringWithFormat:@"Addin device to %@",
self.placeId];
    self.myTF.delegate = self;
    self.myCode.delegate = self;
    self.mybigTF.delegate = self;
    [self.view bringSubviewToFront:self.controlContainerView];
}

- (BOOL)textFieldShouldReturn:(UITextField *)textField {
    [textField resignFirstResponder];
    return NO;
}

- (void)pauseScan {

```

MyUltimateRemote – Jhonatan Tavori

```
}

- (void)resumeScan {
    if (self.keepScanning) {
        NSLog(@"*** RESUMING SCAN!");
        [NSTimer scheduledTimerWithTimeInterval:TIMER_SCAN_INTERVAL target:self
selector:@selector(pauseScan) userInfo:nil repeats:NO];
        [self.centralManager scanForPeripheralsWithServices:nil options:nil];
    }
}

- (void)cleanup {
    [_centralManager cancelPeripheralConnection:self.cc26501];
}

#pragma mark - CBCentralManagerDelegate methods

- (void)centralManagerDidUpdateState:(CBCentralManager *)central {
    BOOL showAlert = YES;
    NSString *state = @"";
    switch ([central state])
    {
        case CBCentralManagerStateUnsupported:
            state = @"This device does not support Bluetooth Low Energy.";
            break;
        case CBCentralManagerStateUnauthorized:
            state = @"This app is not authorized to use Bluetooth Low Energy.";
            break;
        case CBCentralManagerStatePoweredOff:
            state = @"Bluetooth on this device is currently powered off.";
            break;
        case CBCentralManagerStateResetting:
            state = @"The BLE Manager is resetting; a state update is pending.";
            break;
        case CBCentralManagerStatePoweredOn:
            showAlert = NO;
            state = @"Bluetooth LE is turned on and ready for communication.";
            NSLog(@"%@", state);
            self.keepScanning = YES;
            [NSTimer scheduledTimerWithTimeInterval:TIMER_SCAN_INTERVAL target:self
selector:@selector(pauseScan) userInfo:nil repeats:NO];
            [self.centralManager scanForPeripheralsWithServices:nil options:nil];
            break;
        case CBCentralManagerStateUnknown:
            state = @"The state of the BLE Manager is unknown.";
            break;
        default:
            state = @"The state of the BLE Manager is unknown.";
    }

    if (showAlert) {
        UIAlertController *ac = [UIAlertController alertControllerWithTitle:@"Central
Manager State" message:state preferredStyle:UIAlertControllerStyleAlert];
        UIAlertAction *okAction = [UIAlertAction actionWithTitle:@"OK"
style:UIAlertActionStyleCancel handler:nil];
        [ac addAction:okAction];
        [self presentViewController:ac animated:YES completion:nil];
    }
}

- (void)centralManager:(CBCentralManager *)central didDiscoverPeripheral:(CBPeripheral
*)peripheral advertisementData:(NSDictionary *)advertisementData RSSI:(NSNumber *)RSSI {
    // Retrieve the peripheral name from the advertisement data using the
    "kCBAAdvDataLocalName" key
    NSString *peripheralName = [advertisementData objectForKey:@"kCBAAdvDataLocalName"];
    NSLog(@"NEXT PERIPHERAL: %@ (%@)", peripheralName,
peripheral.identifier.UUIDString);
    if (peripheralName) {
```

MyUltimateRemote – Jhonatan Tavori

```
        if ([peripheralName isEqualToString:devName]) {
            self.connected = true;
            self.cc2650l = peripheral;
            self.cc2650l.delegate = self;
            [self.centralManager stopScan];
            [self.centralManager connectPeripheral:self.cc2650l options:nil];
        }
    }
}

- (void)centralManager:(CBCentralManager *)central didConnectPeripheral:(CBPeripheral *)peripheral {

    NSLog(@"**** SUCCESSFULLY CONNECTED TO CC2650 LAUNCHPAD!!!");
    self.searchLab.text = _placeId;
    [peripheral discoverServices:nil];
}

- (void)centralManager:(CBCentralManager *)central
didFailToConnectPeripheral:(CBPeripheral *)peripheral error:(NSError *)error {
    NSLog(@"**** CONNECTION FAILED!");
}

- (void)centralManager:(CBCentralManager *)central didDisconnectPeripheral:(CBPeripheral *)peripheral error:(NSError *)error {
    NSLog(@"**** DISCONNECTED");
}

#pragma mark - CBPeripheralDelegate methods

- (void)peripheral:(CBPeripheral *)peripheral didDiscoverServices:(NSError *)error {
    for (CBService *service in peripheral.services) {
        NSLog(@"We Have Found service: %@", service.UUID);
        [peripheral discoverCharacteristics:nil forService:service];
    }
}

- (void)peripheral:(CBPeripheral *)peripheral
didDiscoverCharacteristicsForService:(CBService *)service error:(NSError *)error {
    for (CBCharacteristic *characteristic in service.characteristics) {
        NSLog(@"We Have Found Characteristic%@", characteristic.UUID);
        [self.cc2650l setNotifyValue:YES forCharacteristic:characteristic];
        if ([characteristic.UUID isEqual:[CBUUID UUIDWithString:UUID_1]]) {
            self.redLed = characteristic;
        }
        if ([characteristic.UUID isEqual:[CBUUID UUIDWithString:UUID_2]]) {
            self.greenLed = characteristic;
        }
    }
}

- (void)peripheral:(CBPeripheral *)peripheral
didUpdateValueForCharacteristic:(CBCharacteristic *)characteristic error:(NSError *)error {
    if (error) {
        NSLog(@"Error changing notification state: %@", [error localizedDescription]);
    } else {
        NSData *dataBytes = characteristic.value;
        NSLog(@"vale for Char is: %@", dataBytes);
    }
}

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender {
}
```

MyUltimateRemote – Jhonatan Tavori

```
-(IBAction)threep:(id)sender {
    NSInteger t = [_myCode.text intValue];
    uint8_t enableValue = (uint8_t) t;
    NSData *enableBytes = [NSData dataWithBytes:&enableValue length:sizeof(uint8_t)];
    [self.cc2650l writeValue:enableBytes forCharacteristic:_redLed
type:CBCharacteristicWriteWithResponse];
    NSLog(@"send %i", enableValue);
}

-(IBAction)fivep:(id)sender {
    NSInteger t = [_myCode.text intValue];
    uint8_t enableValue = (uint8_t) t;
    enableValue = enableValue + 1;
    NSData *enableBytes = [NSData dataWithBytes:&enableValue length:sizeof(uint8_t)];
    [self.cc2650l writeValue:enableBytes forCharacteristic:_redLed
type:CBCharacteristicWriteWithResponse];
    NSLog(@"send %i", enableValue);
}

-(IBAction)switchview:(id)sender {
    [self performSegueWithIdentifier:@"MyDefauktSegueY" sender:self];
}

-(IBAction)search:(id)sender {
    self.centralManager = [[CBCentralManager alloc] initWithDelegate:self queue:nil
options:nil];
    devName = _mybigTF.text;
    self.connected = false;
    self.searchLab.text = @"Searching";
}

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

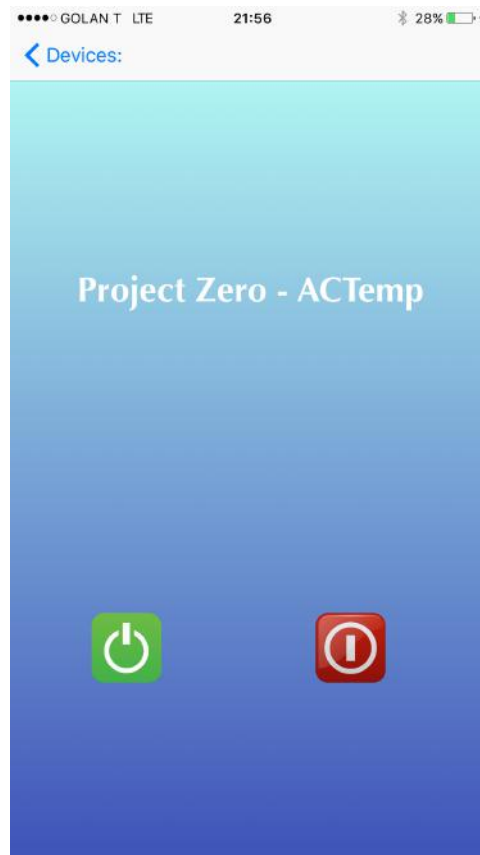
- (IBAction)cancel:(id)sender {
    [self.centralManager cancelPeripheralConnection:_cc2650l];
    [self dismissViewControllerAnimated:YES completion:nil];
}

- (IBAction)save:(id)sender {
    NSString *devdevName = _mybigTF.text;
    NSArray *devarr1 = [NSKeyedUnarchiver unarchiveObjectWithData:[NSUserDefaults
standardUserDefaults] objectForKey:devdevName] ;
    NSMutableArray *devarr2 = [NSMutableArray arrayWithArray:devarr1];
    devName = _myTF.text;
    [devarr2 addObject:devName];
    devarr1 = [devarr2 copy];
    [[NSUserDefaults standardUserDefaults] setObject: [NSKeyedArchiver
archivedDataWithRootObject:devarr1 forKey:devdevName];
    NSInteger t = [_myCode.text intValue];
    [[NSUserDefaults standardUserDefaults] setInteger:t forKey:devName];

    [self.centralManager cancelPeripheralConnection:_cc2650l];
    [self dismissViewControllerAnimated:YES completion:nil];
}

@end
```

הפעלת מכשיר, כולל שידור:



:HomeController.h

```
#import <UIKit/UIKit.h>

@interface HomeController : UIViewController <UITableViewDelegate,
UITableViewDataSource> {

}

@property (strong) NSString *placeid;
@property (strong) NSString *deviceid;
@property (strong) NSString *actid;

@end
```

:HomeController.m

```
#import "HomeController.h"
#import <CoreBluetooth/CoreBluetooth.h>
#import "Constants.h"
```


MyUltimateRemote – Jhonatan Tavori

```
#define TIMER_PAUSE_INTERVAL 0.5
#define TIMER_SCAN_INTERVAL 2.0

@interface HomeViewController () <CBCentralManagerDelegate, CBPeripheralDelegate>

@property (weak, nonatomic) IBOutlet UIView *tableview;

@property (nonatomic, strong) CBCentralManager *centralManager;
@property (nonatomic, strong) CBPeripheral *cc2650l;
@property (nonatomic, strong) CBCharacteristic *redLed;
@property (nonatomic, strong) CBCharacteristic *greenLed;

@property (nonatomic, assign) BOOL keepScanning;
@property (nonatomic, assign) BOOL connected;

-(IBAction)threep:(id)sender;
-(IBAction)fivep:(id)sender;
-(IBAction)switchview:(id)sender;

@end

@implementation HomeViewController {}

- (void)viewDidLoad {
    [super viewDidLoad];

    self.centralManager = [[CBCentralManager alloc] initWithDelegate:self queue:nil
options:nil];
    self.connected = false;
}

- (void)pauseScan {
}

- (void)resumeScan {
    if (self.keepScanning) {
        [NSTimer scheduledTimerWithTimeInterval:TIMER_SCAN_INTERVAL target:self
selector:@selector(pauseScan) userInfo:nil repeats:NO];
        [self.centralManager scanForPeripheralsWithServices:nil options:nil];
    }
}

- (void)cleanup {
    [_centralManager cancelPeripheralConnection:self.cc2650l];
}

#pragma mark - CBCentralManagerDelegate methods

- (void)centralManagerDidUpdateState:(CBCentralManager *)central {
    BOOL showAlert = YES;
    NSString *state = @"";
    switch ([central state])
    {
        case CBCentralManagerStateUnsupported:
            state = @"This device does not support Bluetooth Low Energy.";
            break;
        case CBCentralManagerStateUnauthorized:
            state = @"This app is not authorized to use Bluetooth Low Energy.";
            break;
        case CBCentralManagerStatePoweredOff:
            state = @"Bluetooth on this device is currently powered off.";
            break;
        case CBCentralManagerStateResetting:
            state = @"The BLE Manager is resetting; a state update is pending.";
            break;
    }
}
```

MyUltimateRemote – Jhonatan Tavori

```
        case CBCentralManagerStatePoweredOn:
            showAlert = NO;
            state = @"Bluetooth LE is turned on and ready for communication.";
            NSLog(@"%@", state);
            self.keepScanning = YES;
            [NSTimer scheduledTimerWithTimeInterval:TIMER_SCAN_INTERVAL target:self
selector:@selector(pauseScan) userInfo:nil repeats:NO];
            [self.centralManager scanForPeripheralsWithServices:nil options:nil];
            break;
        case CBCentralManagerStateUnknown:
            state = @"The state of the BLE Manager is unknown.";
            break;
        default:
            state = @"The state of the BLE Manager is unknown.";
    }

    if (showAlert) {
        UIAlertController *ac = [UIAlertController alertControllerWithTitle:@"Central
Manager State" message:state preferredStyle:UIAlertControllerStyleAlert];
        UIAlertAction *okAction = [UIAlertAction actionWithTitle:@"OK"
style:UIAlertActionStyleCancel handler:nil];
        [ac addAction:okAction];
        [self presentViewController:ac animated:YES completion:nil];
    }
}

- (void)centralManager:(CBCentralManager *)central didDiscoverPeripheral:(CBPeripheral
*)peripheral advertisementData:(NSDictionary *)advertisementData RSSI:(NSNumber *)RSSI {
    // Retrieve the peripheral name from the advertisement data using the
    "kCBAAdvDataLocalName" key
    NSString *peripheralName = [advertisementData objectForKey:@"kCBAAdvDataLocalName"];
    NSLog(@"NEXT PERIPHERAL: %@ (%@)", peripheralName,
peripheral.identifier.UUIDString);
    if (peripheralName) {

        if ([peripheralName isEqualToString:_deviceId]) {

            self.connected = true;
            self.cc2650l = peripheral;
            self.cc2650l.delegate = self;
            [self.centralManager stopScan];

            // Request a connection to the peripheral
            [self.centralManager connectPeripheral:self.cc2650l options:nil];
        }
    }
}

- (void)centralManager:(CBCentralManager *)central didConnectPeripheral:(CBPeripheral
*)peripheral {
    [peripheral discoverServices:nil];
}

- (void)centralManager:(CBCentralManager *)central
didFailToConnectPeripheral:(CBPeripheral *)peripheral error:(NSError *)error {
    NSLog(@"**** CONNECTION FAILED!!!");
}

- (void)centralManager:(CBCentralManager *)central didDisconnectPeripheral:(CBPeripheral
*)peripheral error:(NSError *)error {
    NSLog(@"**** DISCONNECTED!");
}

#pragma mark - CBPeripheralDelegate methods

- (void)peripheral:(CBPeripheral *)peripheral didDiscoverServices:(NSError *)error {
    for (CBService *service in peripheral.services) {
```

MyUltimateRemote – Jhonatan Tavori

```
        NSLog(@"We Have Found service: %@", service.UUID);
        [peripheral discoverCharacteristics:nil forService:service];
    }
}

- (void)peripheral:(CBPeripheral *)peripheral
didDiscoverCharacteristicsForService:(CBService *)service error:(NSError *)error {
    for (CBCharacteristic *characteristic in service.characteristics) {
        NSLog(@"We Have Found Characteristic%@", characteristic.UUID);
        [self.cc2650l setNotifyValue:YES forCharacteristic:characteristic];
        if ([characteristic.UUID isEqual:[CBUUID UUIDWithString:UUID_1]]) {
            self.redLed = characteristic;
        }
        if ([characteristic.UUID isEqual:[CBUUID UUIDWithString:UUID_2]]) {
            self.greenLed = characteristic;
        }
    }
}

- (void)peripheral:(CBPeripheral *)peripheral
didUpdateValueForCharacteristic:(CBCharacteristic *)characteristic error:(NSError
*)error {
    if (error) {
        NSLog(@"Error changing notification state: %@", [error localizedDescription]);
    } else {
        NSData *dataBytes = characteristic.value;
        NSLog(@"vale for Char is: %@", dataBytes);
    }
}

-(IBAction)threep:(id)sender {

    NSInteger myint = [[NSUserDefaults standardUserDefaults] integerForKey:_actid];
    uint8_t enableValue = (uint8_t) myint;
    enableValue = enableValue + 2;
    NSData *enableBytes = [NSData dataWithBytes:&enableValue length:sizeof(uint8_t)];
    [self.cc2650l writeValue:enableBytes forCharacteristic:_redLed
type:CBCharacteristicWriteWithResponse];

}

-(IBAction)fivep:(id)sender {

    NSInteger myint = [[NSUserDefaults standardUserDefaults] integerForKey:_actid];
    uint8_t enableValue = (uint8_t) myint;
    enableValue = enableValue + 3;
    NSData *enableBytes = [NSData dataWithBytes:&enableValue length:sizeof(uint8_t)];
    [self.cc2650l writeValue:enableBytes forCharacteristic:_redLed
type:CBCharacteristicWriteWithResponse];

}

- (void)viewWillDisappear:(BOOL)animated {
    [super viewWillDisappear:animated];

    if (self.isMovingFromParentViewController) {
        if (self.cc2650l != nil) {
            [self.centralManager cancelPeripheralConnection:self.cc2650l];
        }
    }
}

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
}
```

MyUltimateRemote – Jhonatan Tavori

@end

: Board ה בצד הקוד

(נכתב בשפת c בcss של TI).

עיקר הקוד נכתב במודול ייעודי שמטפל בתקשורת עם ה-IR.

קובץ ה-IRrt.c:

```
/* XDC module Headers */
#include <xdc/std.h>
#include <xdc/runtime/System.h>
#include <xdc/runtime/Error.h>

/* BIOS module Headers */
#include <ti/sysbios/BIOS.h>
#include <ti/sysbios/knl/Task.h>
#include <ti/sysbios/knl/Clock.h>
#include <ti/sysbios/hal/Hwi.h>

#include "Board.h"
#include "IRapp.h"

#define VEC_MAX                240
#define MIN_PULSES            20

extern PIN_Handle *GetIledPinHandle(void);

PIN_Config IRledPinTable[] = {
    IOID_25      | PIN_GPIO_OUTPUT_EN | PIN_GPIO_LOW  | PIN_PUSHPULL  |
    PIN_DRVSTR_MAX,
    PIN_TERMINATE
};

/*
 * Application button pin configuration table:
 * - Buttons interrupts are configured to trigger on falling edge.
 */
PIN_Config IRbuttonPinTable[] = {
    IOID_12      | PIN_INPUT_EN | PIN_PULLUP | PIN_BM_HYSTERESIS |
    PIN_IRQ_BOTHEDGES      ,
    PIN_TERMINATE
};

/* Pin driver handles */
static PIN_Handle IRbuttonPinHandle;
static PIN_Handle IRledPinHandle , *RedGreenHandle ;

/* Global memory storage for a PIN_Config table */
```



```

{
    while (x--)
    {
        PIN_setOutputValue(IRledPinHandle, IOID_25, LED_1 );
        DelayPulse_0();
        PIN_setOutputValue(IRledPinHandle, IOID_25, LED_1 );
        DelayPulse_1();
    }
}

```

```

Void IR_TaskFunc(uint8_t Pstate)
{
    uint8_t    b , i , *p ;
    uint32_t   x          ;
    p = Off_Const ;
    switch (Pstate)
    {
        case PSTATE_SEND_ON    :
            p = On_Const ;
        case PSTATE_SEND_OFF   :
            Hwi_disable();
            i = 0 ;
            b = 0 ;
            while (b!=0xff)
            {
                b = p[i++] ;
                switch (b)
                {
                    case 0 :
                        MakeSigOn (PLS_0);
                        MakeSigOff(PLS_0);
                        break ;
                    case 1 :
                        MakeSigOn (PLS_0);
                        MakeSigOff(PLS_1);
                        break ;
                    case 2 :
                        MakeSigOn (PLS_2);
                        break ;
                    case 3 :
                        MakeSigOff(PLS_3);
                        break ;
                    case 4 :
                        MakeSigOn (PLS_0) ;
                        MakeSigOff(PLS_4);
                        break ;
                    default :
                        b = 0xff ;
                        break ;
                }
            }
            Hwi_enable();
            Pstate = PSTATE_BREAK ;
    }
}

```

```

        case PSTATE_REC_ON :
(RecIntPoi<(VEC_MAX-1))
RecOn[i+1] = RecInt[i] ;
*RedGreenHandle , Board_LED0 , 1 );
*RedGreenHandle , Board_LED1 , 0 );
        case PSTATE_REC_OFF :
(RecIntPoi<(VEC_MAX-1))
RecOff[i+1] = RecInt[i] ;
*RedGreenHandle , Board_LED0 , 0 );
*RedGreenHandle , Board_LED1 , 1 );
        case PSTATE_SEND_REC_OFF:
        case PSTATE_SEND_REC_ON :
PSnd = RecOn ;
PSnd = RecOff ;
        if (Pstate == PSTATE_SEND_REC_ON)
        if (PSnd[0]>MIN_PULSES)
        {
            Hwi_disable();
            for (i=1; i<PSnd[0]; i++)
            {
                x = PSnd[i] ;
                x *= 113 ;
                x /= 100 ;
                if (i & 0x01) MakeSigOn
                else
                }
            MakeSigOff(100);
            Hwi_enable();
        }
        break ;
        case MSTATE_IDLE :
        default :
        break ;
    }
    PIN_setOutputValue(IRledPinHandle, IOID_25, LED_1);

```

```
}

Void IRclk0Fxn(UArg arg0)
{
    static uint8_t Tm50 ;
    Global_IntTm++;
    Tm50++;
    if (Tm50>31) // 1ms
    {
        Tm50 = 0 ;
        if (Global_msTm) Global_msTm-- ;
    }
}

void IRbuttonCallbackFxn(PIN_Handle handle, PIN_Id pinId)
{
    if (Global_msTm==0)
    {
        RecIntPoi    = 0 ;
    }
    else
    {
        if ((RecIntPoi<VEC_MAX) && (Global_IntTm < 1000))
        {
            RecInt[RecIntPoi++] = Global_IntTm ;
        }
    }
    Global_IntTm = 0 ;
    Global_msTm  = 400 ;
}

void IRmain(void)
{
    Clock_Params clkParams ;
    Error_Block  eb        ;
    RedGreenHandle    = GetIledPinHandle() ;
    IRledPinHandle    = PIN_open(&IRledPinState , IRledPinTable );
    IRbuttonPinHandle = PIN_open(&IRbuttonPinState, IRbuttonPinTable);
    PIN_registerIntCb(IRbuttonPinHandle, &IRbuttonCallbackFxn);

    Error_init(&eb);
    Clock_Params_init(&clkParams);
    clkParams.period    = 3    /* 3 Clock ticks */
    clkParams.startFlag = TRUE /* start immediately */
    Clock_create((Clock_FuncPtr)IRclk0Fxn, 4, &clkParams, &eb);
}
}
```

: IRapp.h קובץ

```
#ifndef IRAPP_IRAPP_H_
#define IRAPP_IRAPP_H_

#define PSTATE_IDLE 0
#define PSTATE_SEND_ON 1
#define PSTATE_SEND_OFF 2
#define PSTATE_REC_ON 3
#define PSTATE_REC_OFF 4
#define PSTATE_SEND_REC_ON 5
#define PSTATE_SEND_REC_OFF 6
#define PSTATE_BREAK 99

#define MSTATE_IDLE 0
#define MSTATE_REC_ON 1
#define MSTATE_REC_OFF 2

#define LED_0 1
#define LED_1 0
#define TM_PLS 200

#define DLY_LOOP_0 28
#define DLY_LOOP_1 44
#define PLS_0 23 // 30
#define PLS_1 66 // 60
#define PLS_2 340
#define PLS_3 170
#define PLS_4 750

#endif /* IRAPP_IRAPP_H_ */
```