# PROJECT IOT102

# Tên project “Đồng hồ kĩ thuật số”

|  |  |  |
| --- | --- | --- |
| Nhóm 2 | Vũ Thanh Lâm | HE160328 |
| Nhóm 2 | Nguyễn Đức Minh | HE160265 |
| Nhóm 2 | Nguyễn Ngọc Anh | HE153033 |

## I. Ý tưởng

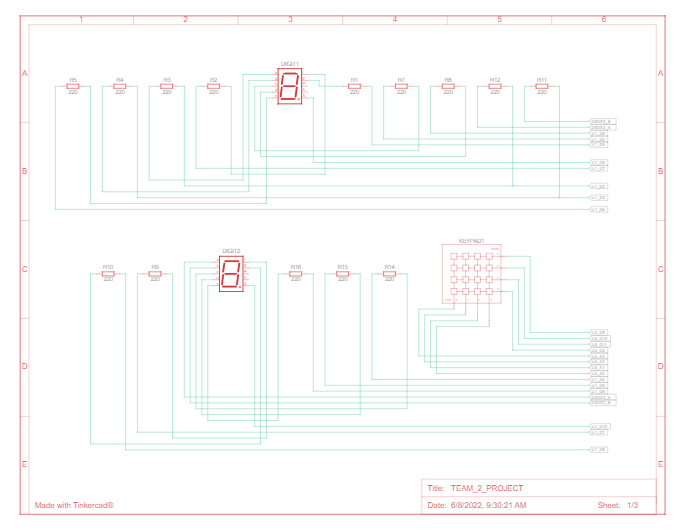
Thiết kế và lắp ráp một chiếc đồng hồ kĩ thuật số có khả năng cài đặt, hiển thị thời gian và hẹn báo thức

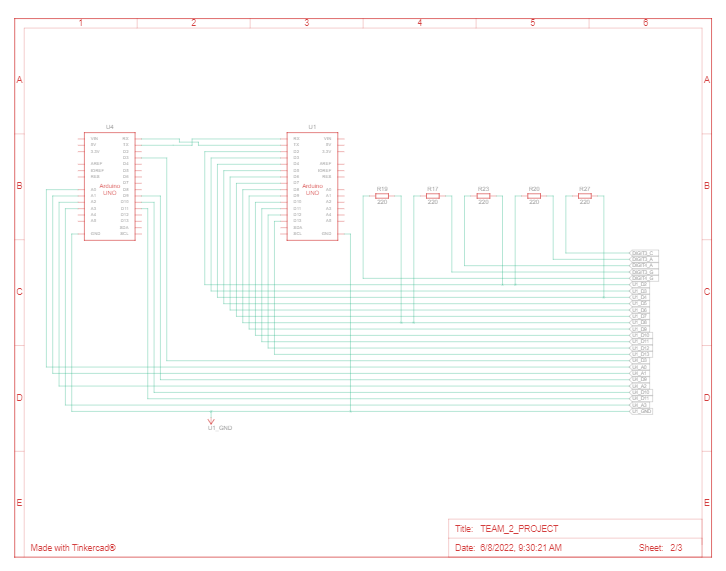
## II. Nội dung

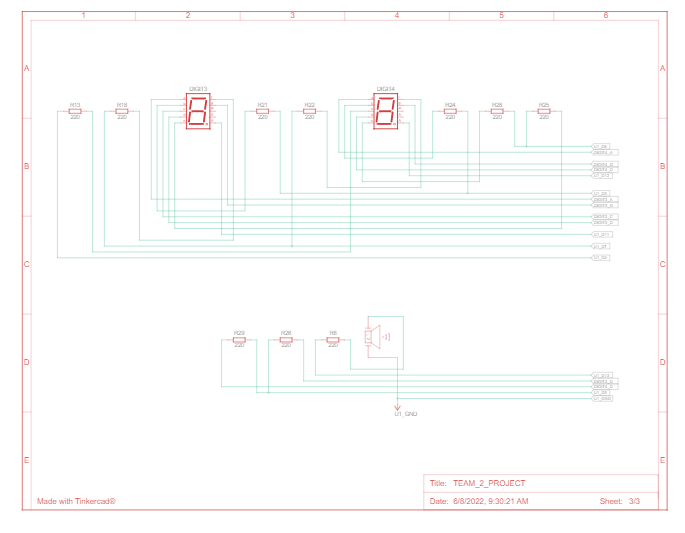
### 1. Linh kiện (Hardware Required

* Adruino Uno R3
* Anode 7 Segment Display
* 220 Ω Resistor
* Keypad 4x4
* Buzzer

### 2. Mạch nguyên lý (Schematic)

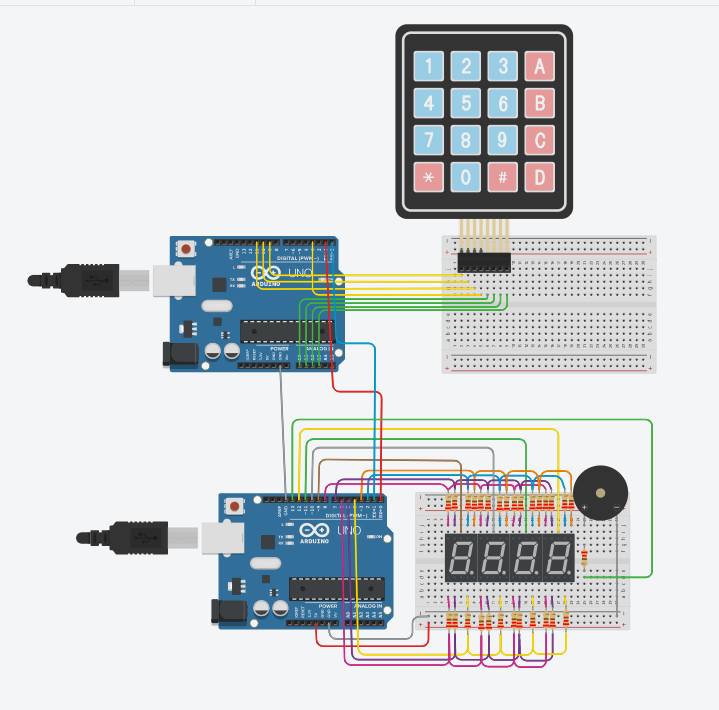






### 3. Mạch kết nối (Circuit )

Mạch mô phỏng hoặc mạch thực tế.

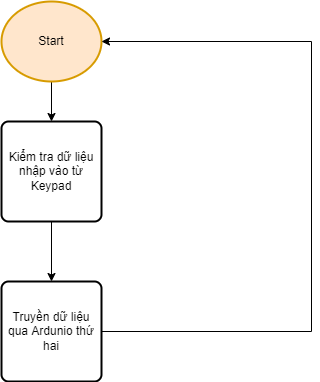


### 4. Lưu đồ thuật toán (Flow chart)

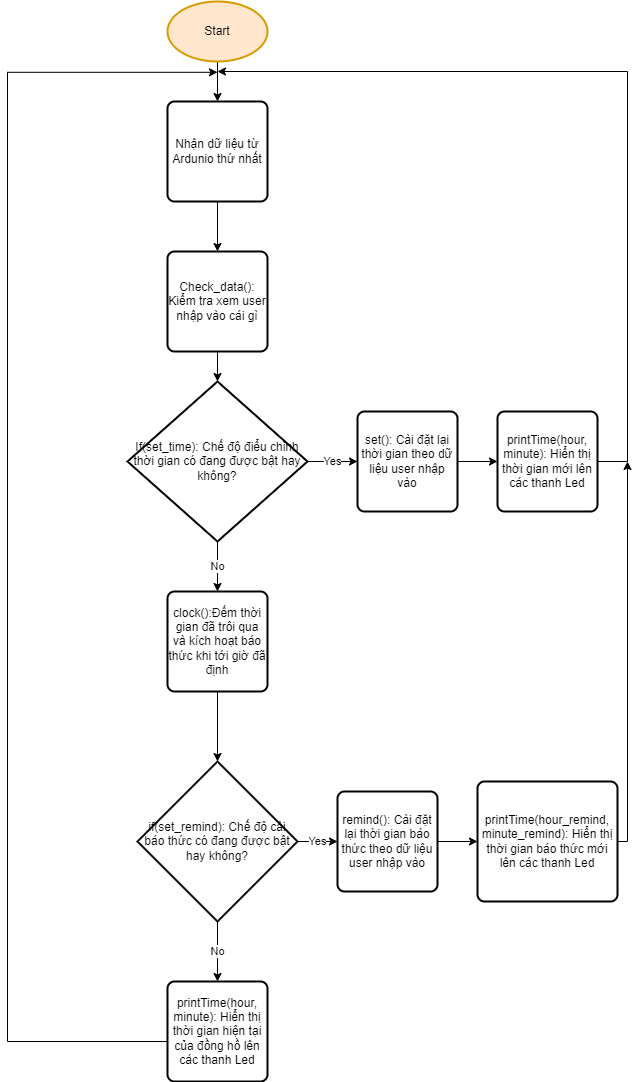
Lưu đồ giải thuật của hàm **void loop**

Tham khảo công cụ vẽ: https://app.diagrams.net/

Ardunio thứ nhất



Ardunio thứ hai



## III. Kết quả đạt được

Đã hoàn thành tất cả các chức năng so với ý tưởng đã đưa ra

## IV. Phụ lục

### 1. Link Tinkercad

* Link mô phỏng: <https://www.tinkercad.com/things/4Z0ZajSkh4o-team2project/editel?sharecode=U6UJ3j1cYMw9eW3B4spYo6xcrKN9wsAnLTJFq2ujm1E&fbclid=IwAR3BqnZqsOXEcgiswz0mZfFawL6z1GKW2-WeX3Yi09i7lDwFCSMaHEvikEg>
* Link video

### 2. Code

Ardunio thứ nhất (Gắn trực tiếp với Keypad)

#include<stdio.h>

#include<stdlib.h>

#include <Keypad.h>

const byte ROWS = 4;

const byte COLS = 4;

char hexaKeys[ROWS][COLS] = {

{'1', '2', '3', 'A'},

{'4', '5', '6', 'B'},

{'7', '8', '9', 'C'},

{'\*', '0', '#', 'D'}

};

byte rowPins[ROWS] = {9, 10, 11, 3};

byte colPins[COLS] = {A0, A1, A2, A3};

Keypad customKeypad = Keypad(makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);

int set\_time = 0;

int set\_remind = 0;

int set\_hour = 0;

int set\_minute = 0;

void setup() {

Serial.begin(9600);

}

void loop() {

char customKey = customKeypad.getKey();

//turn on/off mode remind

if(customKey == '\*'){

Serial.write('\*');

return;

}

//press 'A' to set running time

if(customKey == 'A' && set\_remind == 0) {

set\_time = !set\_time;

Serial.write('A');}

if(set\_time && customKey != NULL && customKey !='A')

{

Serial.write(customKey);

}

//press 'D' to set remind time

if(customKey == 'D' && set\_time==0) {

set\_remind = !set\_remind;

Serial.write('D');}

if(set\_remind && customKey != NULL && customKey !='D')

{

Serial.write(customKey);

}

}

Ardunio thứ hai:

#define segA 2//connecting segment A to PIN2

#define segB 3// connecting segment B to PIN3

#define segC 4// connecting segment C to PIN4

#define segD 5// connecting segment D to PIN5

#define segE 6// connecting segment E to PIN6

#define segF 7// connecting segment F to PIN7

#define segG 8// connecting segment G to PIN8

#define ledH1 9

#define ledH2 10

#define ledM1 11

#define ledM2 12

int count=0;//count integer for 0-9 increment

int time = 10;

int hour = 9;

int minute = 0;

int set\_remind = 0;

int hour\_remind =0;

int minute\_remind = 0;

int set\_time = 0;

int set\_hour=0;

int set\_minute=0;

char customKey;

int on\_buzzer=0;

int remind\_mode = 0;

void setup()

{

for (int i=1;i<=13;i++)

{

pinMode(i, OUTPUT);

}

Serial.begin(9600);

}

//each cycle take 40 miliseconds

void loop()

{

//check data to activate function

check\_data();

//set time

if(set\_time) {

set();

printTime(hour, minute);

}

//remind

else {

//run clock

clock();

if(set\_remind) {

remind();

printTime(hour\_remind, minute\_remind);

return;

}

printTime(hour, minute);

}

}

void check\_data(){

if (Serial.available()){

customKey = Serial.read();

if(customKey =='A' && set\_remind == 0) set\_time = !set\_time;

if(customKey =='B' && set\_minute == 0) set\_hour = !set\_hour;

if(customKey =='C' && set\_hour == 0) set\_minute = !set\_minute;

if(customKey =='D' && set\_time == 0) set\_remind = !set\_remind;

if(customKey =='\*') {

remind\_mode = !remind\_mode;

if(on\_buzzer) {

buzzer();

}

}

if(customKey =='#' && set\_time ==1){

hour = 0;

minute = 0;

}

if(customKey =='#' && set\_remind ==1){

hour\_remind = 0;

minute\_remind = 0;

}

}

else customKey = 'N';

if(customKey !='N' && customKey != NULL)

{Serial.println(customKey);

Serial.println(hour);

}

}

void remind(){

int value = int(customKey)-48;

if(value > 9 || value <0) {

return;

}

if(set\_hour){

hour\_remind = hour\_remind\*10+value;

hour\_remind = check\_time(hour\_remind,0, 23);

}

if(set\_minute){

minute\_remind = minute\_remind\*10+value;

minute\_remind = check\_time(minute\_remind,0,59);

}

}

void set(){

int value = int(customKey)-48;

if(value > 9 || value <0) {

return;

}

if(set\_hour){

hour = hour\*10+value;

hour = check\_time(hour,0, 23);

}

if(set\_minute){

minute = minute\*10+value;

minute = check\_time(minute,0,59);

}

}

int check\_time(int i,int min, int max){

if(i > max) i = min;

//if(i < min) i = min;

return i;

}

void clock(){

if (hour == hour\_remind && minute == minute\_remind && count == 0 && remind\_mode){

buzzer();

}

if(count==1500){

if(minute == 59){

if(hour == 23) hour =0;

else hour++;

minute = 0;

}else minute++;

count = 0;

}else count++;

if(on\_buzzer && count == 150 ) buzzer();

}

void buzzer(){

on\_buzzer = !on\_buzzer;

if(on\_buzzer) digitalWrite(13,1);

else {

digitalWrite(13,0);

}

}

void printTime(int hour, int minute){

//print H1

digitalWrite(ledH1, HIGH);

digitalWrite(ledH2, LOW);

digitalWrite(ledM1, LOW);

digitalWrite(ledM2, LOW);

print(hour/10);

delay(time);

//print h2

digitalWrite(ledH1, LOW);

digitalWrite(ledH2, HIGH);

digitalWrite(ledM1, LOW);

digitalWrite(ledM2, LOW);

print(hour%10);

delay(time);

//print m1

digitalWrite(ledH1, LOW);

digitalWrite(ledH2, LOW);

digitalWrite(ledM1, HIGH);

digitalWrite(ledM2, LOW);

print(minute/10);

delay(time);

//print m2

digitalWrite(ledH1, LOW);

digitalWrite(ledH2, LOW);

digitalWrite(ledM1, LOW);

digitalWrite(ledM2, HIGH);

print(minute%10);

delay(time);

}

void print(int i){

switch (i)

{

case 0://when count value is zero show”0” on disp

digitalWrite(segA, LOW);

digitalWrite(segB, LOW);

digitalWrite(segC, LOW);

digitalWrite(segD, LOW);

digitalWrite(segE, LOW);

digitalWrite(segF, LOW);

digitalWrite(segG, HIGH);

break;

case 1:// when count value is 1 show”1” on disp

digitalWrite(segA, HIGH);

digitalWrite(segB, LOW);

digitalWrite(segC, LOW);

digitalWrite(segD, HIGH);

digitalWrite(segE, HIGH);

digitalWrite(segF, HIGH);

digitalWrite(segG, HIGH);

break;

case 2:// when count value is 2 show”2” on disp

digitalWrite(segA, LOW);

digitalWrite(segB, LOW);

digitalWrite(segC, HIGH);

digitalWrite(segD, LOW);

digitalWrite(segE, LOW);

digitalWrite(segF, HIGH);

digitalWrite(segG, LOW);

break;

case 3:// when count value is 3 show”3” on disp

digitalWrite(segA, LOW);

digitalWrite(segB, LOW);

digitalWrite(segC, LOW);

digitalWrite(segD, LOW);

digitalWrite(segE, HIGH);

digitalWrite(segF, HIGH);

digitalWrite(segG, LOW);

break;

case 4:// when count value is 4 show”4” on disp

digitalWrite(segA, HIGH);

digitalWrite(segB, LOW);

digitalWrite(segC, LOW);

digitalWrite(segD, HIGH);

digitalWrite(segE, HIGH);

digitalWrite(segF, LOW);

digitalWrite(segG, LOW);

break;

case 5:// when count value is 5 show”5” on disp

digitalWrite(segA, LOW);

digitalWrite(segB, HIGH);

digitalWrite(segC, LOW);

digitalWrite(segD, LOW);

digitalWrite(segE, HIGH);

digitalWrite(segF, LOW);

digitalWrite(segG, LOW);

break;

case 6:// when count value is 6 show”6” on disp

digitalWrite(segA, LOW);

digitalWrite(segB, HIGH);

digitalWrite(segC, LOW);

digitalWrite(segD, LOW);

digitalWrite(segE, LOW);

digitalWrite(segF, LOW);

digitalWrite(segG, LOW);

break;

case 7:// when count value is 7 show”7” on disp

digitalWrite(segA, LOW);

digitalWrite(segB, LOW);

digitalWrite(segC, LOW);

digitalWrite(segD, HIGH);

digitalWrite(segE, HIGH);

digitalWrite(segF, HIGH);

digitalWrite(segG, HIGH);

break;

case 8:// when count value is 8 show”8” on disp

digitalWrite(segA, LOW);

digitalWrite(segB, LOW);

digitalWrite(segC, LOW);

digitalWrite(segD, LOW);

digitalWrite(segE, LOW);

digitalWrite(segF, LOW);

digitalWrite(segG, LOW);

break;

case 9:// when count value is 9 show”9” on disp

digitalWrite(segA, LOW);

digitalWrite(segB, LOW);

digitalWrite(segC, LOW);

digitalWrite(segD, LOW);

digitalWrite(segE, HIGH);

digitalWrite(segF, LOW);

digitalWrite(segG, LOW);

break;

break;

}

}