

PERSONAL TELEVISION - 2



What is our GOAL for this CLASS?

In this class, we learned about Arduino Mega boards. We connected a PAL TV component and IR remote sensor to the Mega board.

What did we ACHIEVE in the class TODAY?

- Connected IR Remote and PAL TV to the Arduino Mega board
- Controlled the TV using the remote
- Change channels on TV using the remote

Which CONCEPTS/ CODING BLOCKS did we cover today?

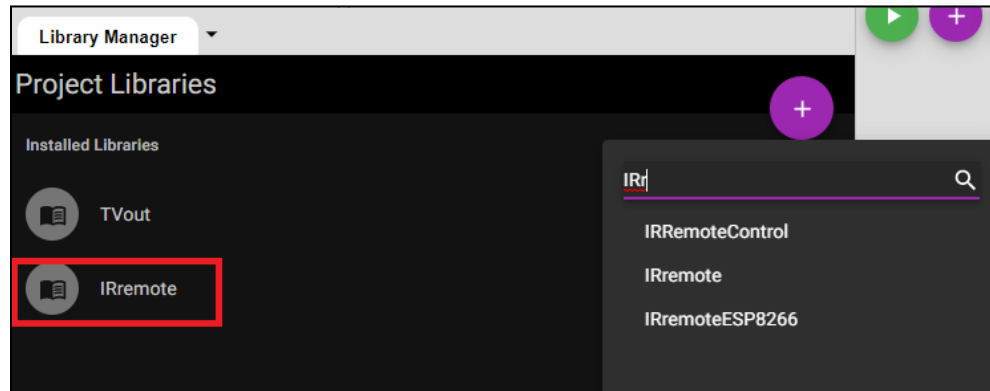
- Concepts : Arduino Mega board, Pin connections of Mega board
- Coding blocks : IRReceiver.h, TVout.h, if blocks, methods and method calls

How did we DO the activities?

1. Learned about the memory limitation of Arduino UNO board:

We connected a PAL TV and a remote to the Arduino UNO board. Then we imported the library called IRremote.h to control the TV using the remote, but the microcontroller crashed due to the heavy load and no output was displayed.

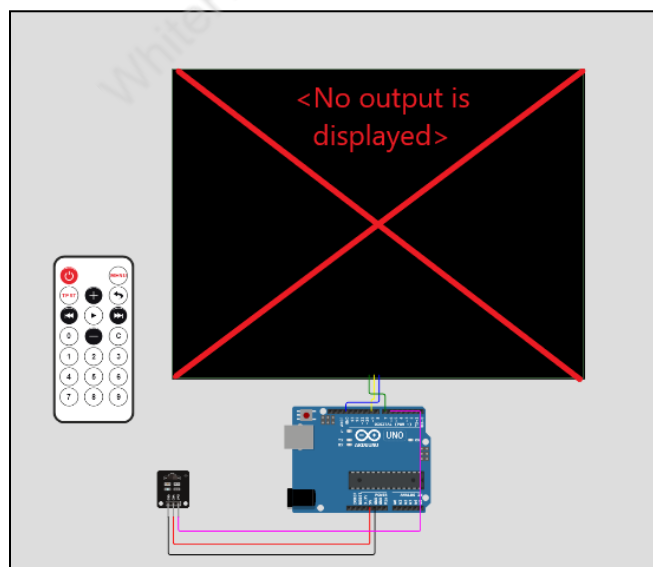
- import IRremote library



- include the header file

```
#include <TVout.h>
#include "a.h"
#include "e.h"
#include "m.h"
#include "o.h"
#include "u.h"
#include "disturbance1.h"
#include "disturbance2.h"
#include "font6x8.h"
#include "IRremote.h"
```

- No output was displayed.



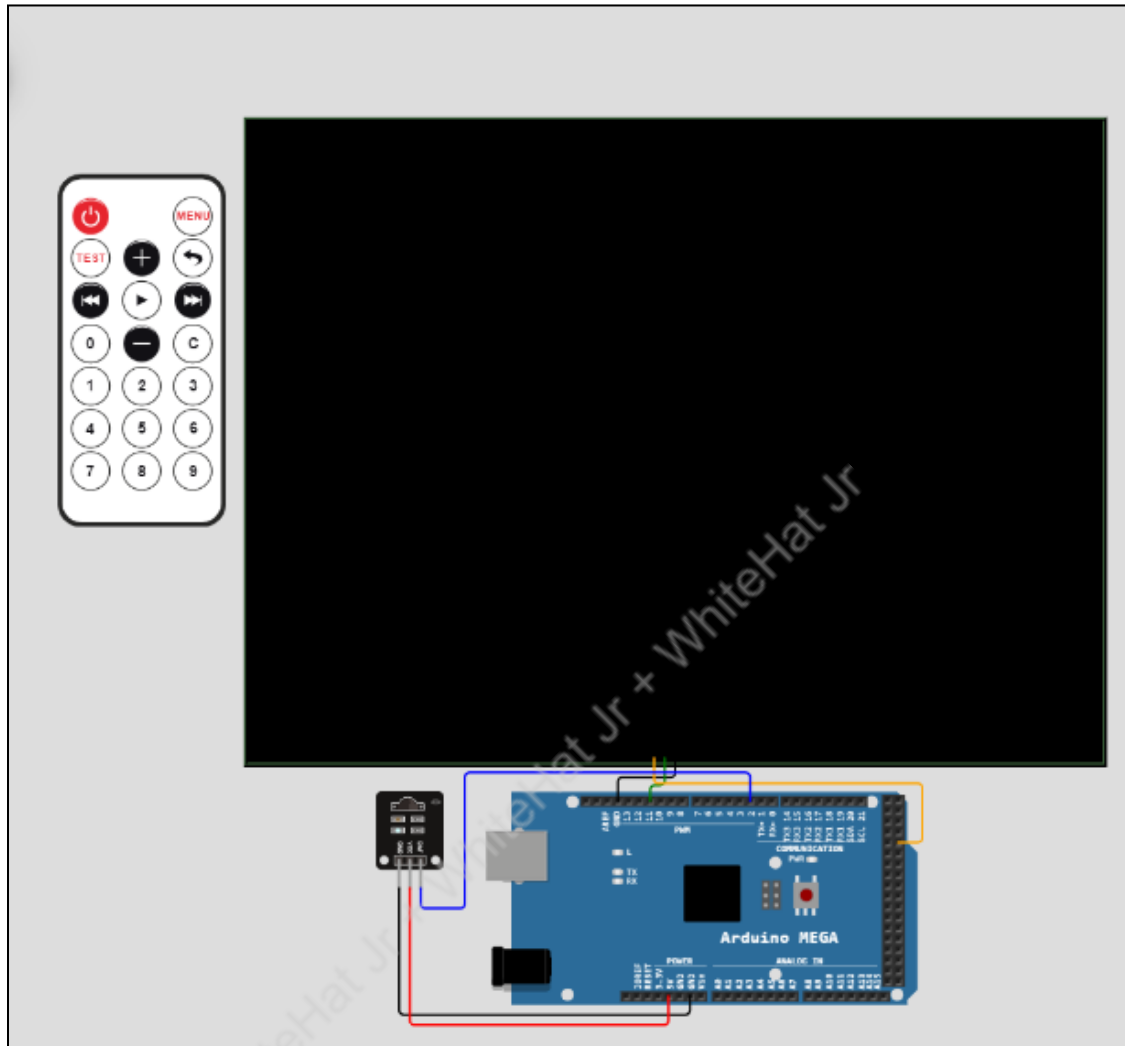
2. Learned about the **Arduino Mega board**:

The Arduino Mega board has better memory capacity compare to that of UNO

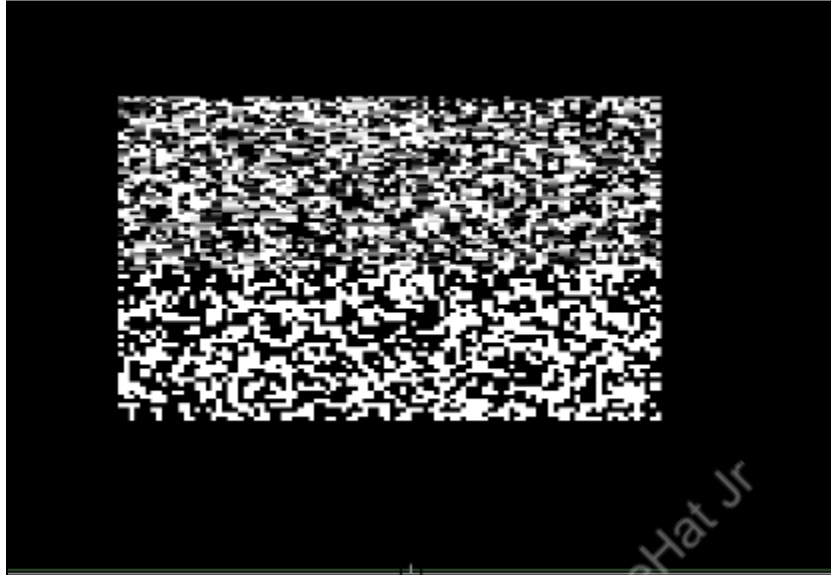
Memory	Mega	UNO
Flash memory	256KB	32KB
EEPROM	4KB	1KB
SRAM	8KB	2KB

3. Used an Arduino Mega board to connect the TV and remote**PIN connections of Mega**

IR Receiver	Arduino Mega PIN
VCC	5V
DAT	2
GND	GND3
PAL TV	Arduino Mega PIN
VIDEO	D29
SYNC	11
GND	GND1



OUTPUT :



4. Control the TV using remote

- TV can be switched ON and OFF
- Channel can be set to News or Dance

5. Coding steps to control the TV

- a. Create an object of the receiver class.

```
#include "font6x8.h"
#include <IRremote.h>

TVout tv;

IRrecv receiver(2);
```

- b. Comment out a few lines in the **setup()** method that calls the **starting_animation()** and **speaking_animation()**.

```
void setup(){  
  
    tv.select_font(font6x8);  
    Serial.begin(9600);  
  
    // tv.begin(PAL);  
    // tv.clear_screen();  
    // tv.delay(100);  
    // starting_animation();  
    // tv.clear_screen();  
    // tv.delay(100);  
    // speaking_animation(35,0);  
    // tv.clear_screen();  
  
}
```

- c. Two methods have been defined in the code, **enable_receiver()** and **disable_receiver()**.
- **enable_receiver()** : enables the receiver and disables the display on the TV by calling **tv.end()**
 - **disable_receiver()** : disables the receiver and enables the display on the TV by calling **tv.begin()**

```
void disable_receiver(){  
    receiver.disableIRIn(); // disable receiver  
    tv.begin(PAL); //enable tv  
}  
  
void enable_receiver(){  
    tv.end(); // disable tv  
    receiver.enableIRIn(); // enable receiver  
}
```

- d. Call **enable_receiver()** from the **setup()** method.

```
void setup(){
  Serial.begin(9600);
  tv.select_font(font6x8); // selecting font

  enable_receiver();

  // tv.begin(PAL);
  // tv.clear_screen();
  // tv.delay(100);
  // starting_animation();
  // tv.clear_screen();
  // tv.delay(100);
  // speaking_animation(35,0);
  // tv.clear_screen();
}
```

- e. Declare two variables **tv_state** and **last_tv_state** at the beginning of the file. They are both initialized to zero which indicates that the TV is currently switched off.

```
TVout tv;
IRrecv receiver(2);

int tv_state = 0, last_tv_state = 0; // initially off
```

- f. The **loop()** method runs continuously. Inside the **loop()** method we will do the following
1. Check if an IR signal is received.
 2. If so, store the signal in a variable called **response**.
 3. If the response value is 162, 162 stands for power button on the remote
 - a. Flip the state of the TV . The TV has two states :
 1. TV is just switched on,
 2. TV is in channel selection mode.
 4. Call the **television()** method
 5. Resume receiving signals.

```
void loop(){
    if (receiver.decode()){
        int response = receiver.decodedIRData.command;
        if (response == 162)tv_state = !tv_state; // flip the state
        television(response);
        receiver.resume();
    }
}
```

- g. The **select_channel()** method displays the two channel options.

```
void select_channel(){
    tv.println(30,40, "SELECT A CHANNEL");
    tv.println(45,50,"0 : NEWS");
    tv.println(45,60,"1 : DANCE");
    tv.delay_frame(3);
}
```

- h. Define the **television()** method..

The method accepts one parameter, **button_pressed** which holds the value of the button pressed on the remote control.

- Before starting any display on the TV, we will disable the receiver and call **tv.begin()**
- Check if TV is on
 - If so check if it was switched on just now, this decides whether the starting animation needs to be shown or not
 - Show the starting animation
 - Show the channel selection
 - Change the value of variable **last_tv_state**
 - If not, that is if the channel selection screen is displayed then,
 - If button pressed is 0, show news channel
 - If button pressed is 1, show dnce channel
- else if TV is switched off,
 - Clear the TV screen
 - Bring a small delay
 - Change the value of variable **last_tv_state**
- As a final step, enable the receiver and call **tv.end()**


```
void television(int code){
    disable_receiver();
    if (tv_state){ // tv is on
        if (last_tv_state == 0){ // tv is just turned on
            starting_animation();
            select_channel();
            last_tv_state = tv_state;
        }
        else if (last_tv_state == 1){
            if (code == 104)news_channel();
            else if (code == 48)dance_channel();
        }
    }
    else{ // tv is off
        tv.clear_screen();
        tv.delay(100);
        last_tv_state = tv_state;
    }
    enable_receiver();
}
```

[Click here](#) to view the reference video.

What's NEXT?

In the **next class**, we will learn about robots. We will use a tool called Webots to build a Robot.

Expand Your Knowledge

To know more about **Arduino Mega board**, [click here](#).