

Topic	PERSONAL TELEVISION - 2	
Class Description	Students will learn about Arduino Mega boards. Students will connect a PAL TV component and IR remote sensor to the Mega board.	
Class	PRO C277	
Class time	50 mins	
Goal	<ul style="list-style-type: none"> • Learn about Arduino Mega boards. • Connect IR Remote and PAL TV to the Mega board • Control the TV using the remote • Change channels on TV using the remote 	
Resources Required	<ul style="list-style-type: none"> • Teacher Resources: <ul style="list-style-type: none"> ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen ○ Smartphone • Student Resources: <ul style="list-style-type: none"> ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen 	
Class structure	Warm-Up Teacher-Led Activity Student-Led Activity Wrap-Up	10 mins 15 mins 15 mins 10 mins
WARM-UP SESSION - 10 mins		
Teacher Action		Student Action
Hello <student's name>. How are you? It's great to see you! Hope you had a great day so far. Following are the WARM-UP session deliverables: <ul style="list-style-type: none"> • Greet the student. • Revision of previous class activities. • Quizzes. 		ESR: Varied. Click on the slide show tab and present the slides

WARM-UP QUIZ
 Click on In-Class Quiz

Activity Details

Following are the session deliverables:

- Appreciate the student.
- Narrate the story by using hand gestures and voice modulation methods to bring in more interest in students.

TEACHER-LED ACTIVITY 15mins

Teacher Initiates Screen Share

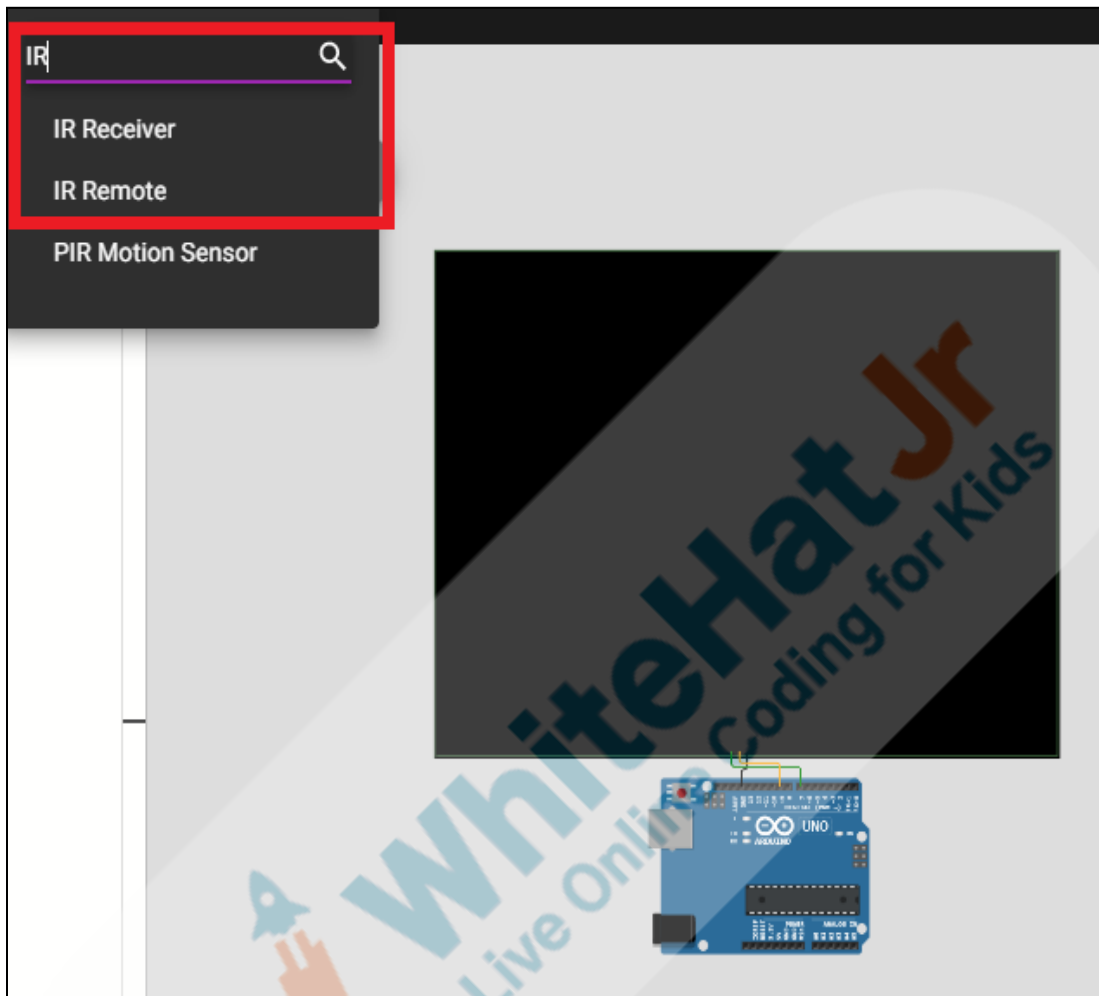
- Understand about Mega boards.
- Use Mega board to connect PAL TV and remote sensor in the circuit
- Write code switch on and off the TV using the power button of the remote

Teacher Action	Student Action
Do you remember what we did in the previous class?	ESR: We connected a PAL TV to an Arduino UNO board. We learned how to display images and text on the PAL TV.
Do you remember what we discussed about displaying images as bitmaps?	ESR: A bitmap is the representation of the image in 0s and 1s . By using a combination of black and white we can draw images on a screen.

<p>That's wonderful, you remember the concepts of the previous class very well.</p> <p><i>If the student has any doubts, clarify the doubts.</i></p>	
<p>Remember how we connected an IR Remote to the Aurino board in the class before?</p> <p>Now let's connect both the TV and remote to the same Arduino board and see what happens.</p>	<p>ESR: Yes</p>
<p>Let us add the IR Remote and IR Receiver to the circuit of the last class, the one that contains the PAL TV.</p> <p>Excellent!</p> <p>Let us open the previous class circuit and add the IRreceiver, remote control and the remote sensor to the circuit of PAL TV.</p> <p><i>Teacher opens the files of Teacher Activity 2 in wokwi simulator and starts adding the components.</i></p> <p><i>Note to teacher : Add the components alone, do not add the library IRremote.h in the code.</i></p>	
<p>Can you tell me the components to be added?</p> <p>That's correct, let's add the two components to the circuit.</p>	<p>ESR: IR Remote and IR Receiver</p>
<p>Click on the Plus button</p>	



Add the two components : **IR Receiver** and **IR Remote**

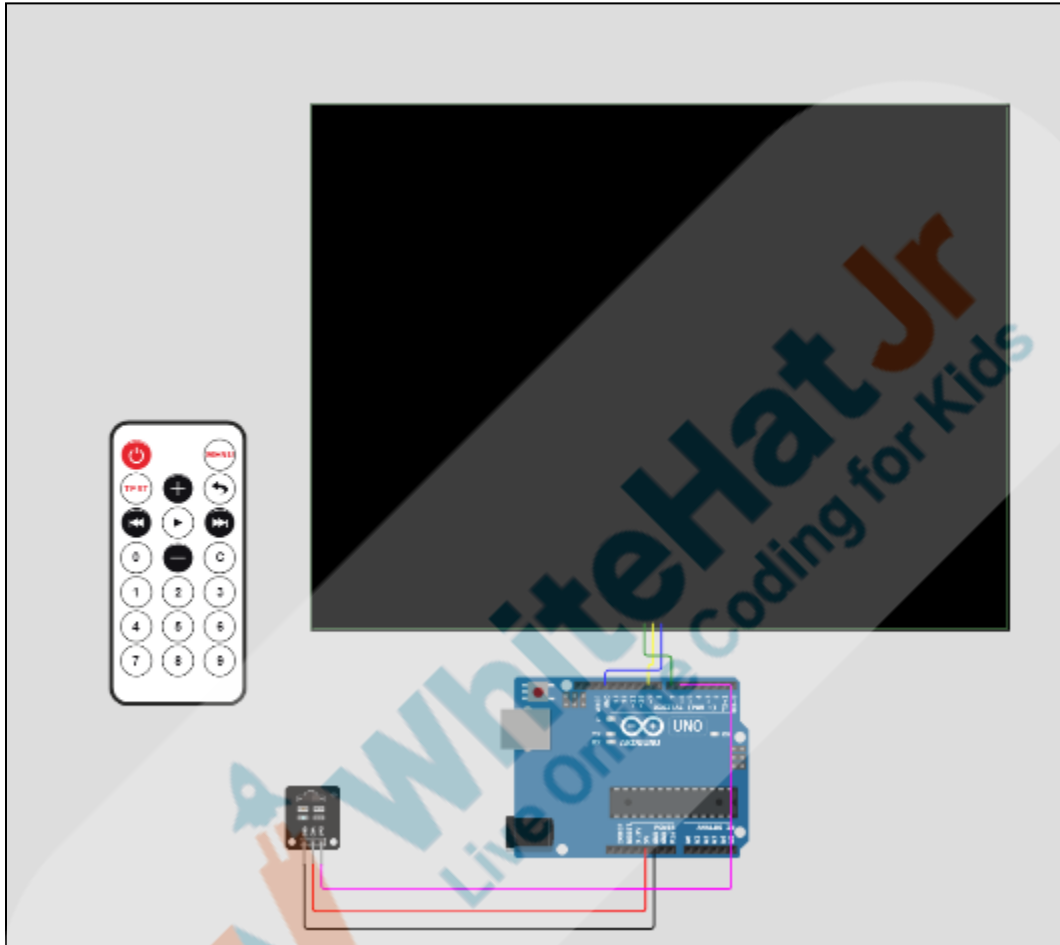


Now let's complete the connections.

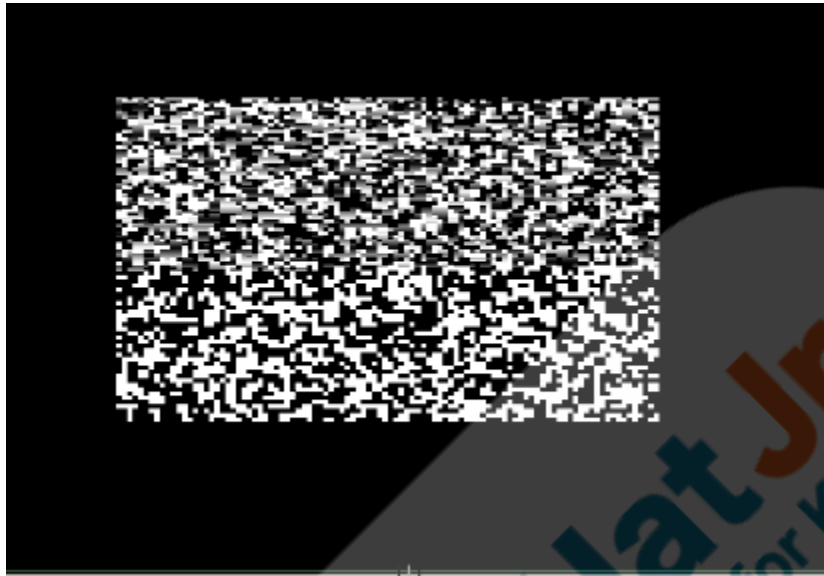
1. The PAL TV is already connected
2. The IRReceiver will be connected in this way:

IR Receiver	Arduino Uno PIN
VCC	5V
GND	GND.2
DAT	D6

The circuit would like this after the connections are made.



OUTPUT:



Now it is time to add the library for the remote control.

Can you tell me the libraries that we need to add once we add the components?

Let us add the library and include the header file in **sketch.ino**

ESR: IRreceiver.h

Library Manager

Project Libraries

Installed Libraries

TVout

IRremote

IRr

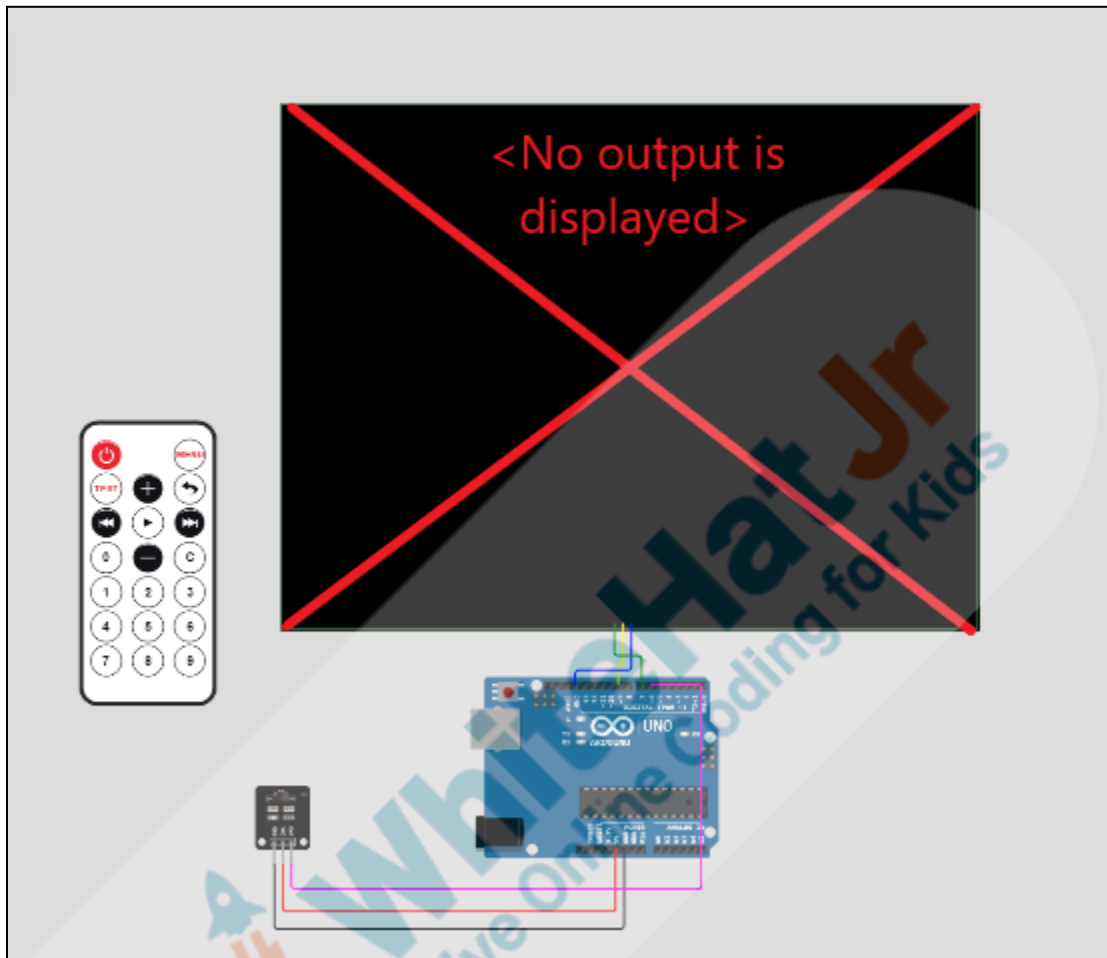
IRRemoteControl

IRremote

IRremoteESP8266

```
#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"
```

Once the library is added and the header file is included in sketch.ino, let's run the simulation to see what happens.



As you can see, no output is displayed.

We know that the code is correct and connections are perfect, so why could this be happening?

The Arduino UNO board that we have in the circuit has certain memory limitations. The **IRremote.h** is a large file with a lot of functions in it. So as soon as we added the **IRremote.h** library, the UNO could not handle it and crashed.

ESR: Varied.

So to make the circuit working again. We will have to add a

much more powerful board called the Arduino Mega board.

When we start a new project on [wokwi](#), we can select the type of processor we want. By selecting Arduino Mega, the Mega board gets added.

The Mega board has better memory capacity compared to the UNO board. Hence projects that have more components or a lot of code may require a Mega board.

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



Let me open the project in which the circuit is complete with the Arduino Mega board.

Teacher opens [Teacher Activity 2](#).

As you can see the code in the project is the same as before. However, the circuit has changed. Can you look closely at the circuit and tell what the major change is?

ESR: The board in the circuit is an Arduino Mega board.

Yes, the microcontroller board is the Arduino Mega board with 54 digital input or output pins based on ATmega2560. 14 of the 54 digital pins are used as PWM outputs and there are 16 analog inputs, 4 UARTS, crystal oscillator, USB connection, a power jack, and a reset button. The Mega board is slightly bigger in size than UNO, but both have the same clock speed of 16MHz.

The connections made on the Arduino Mega board are slightly different from the UNO board.

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

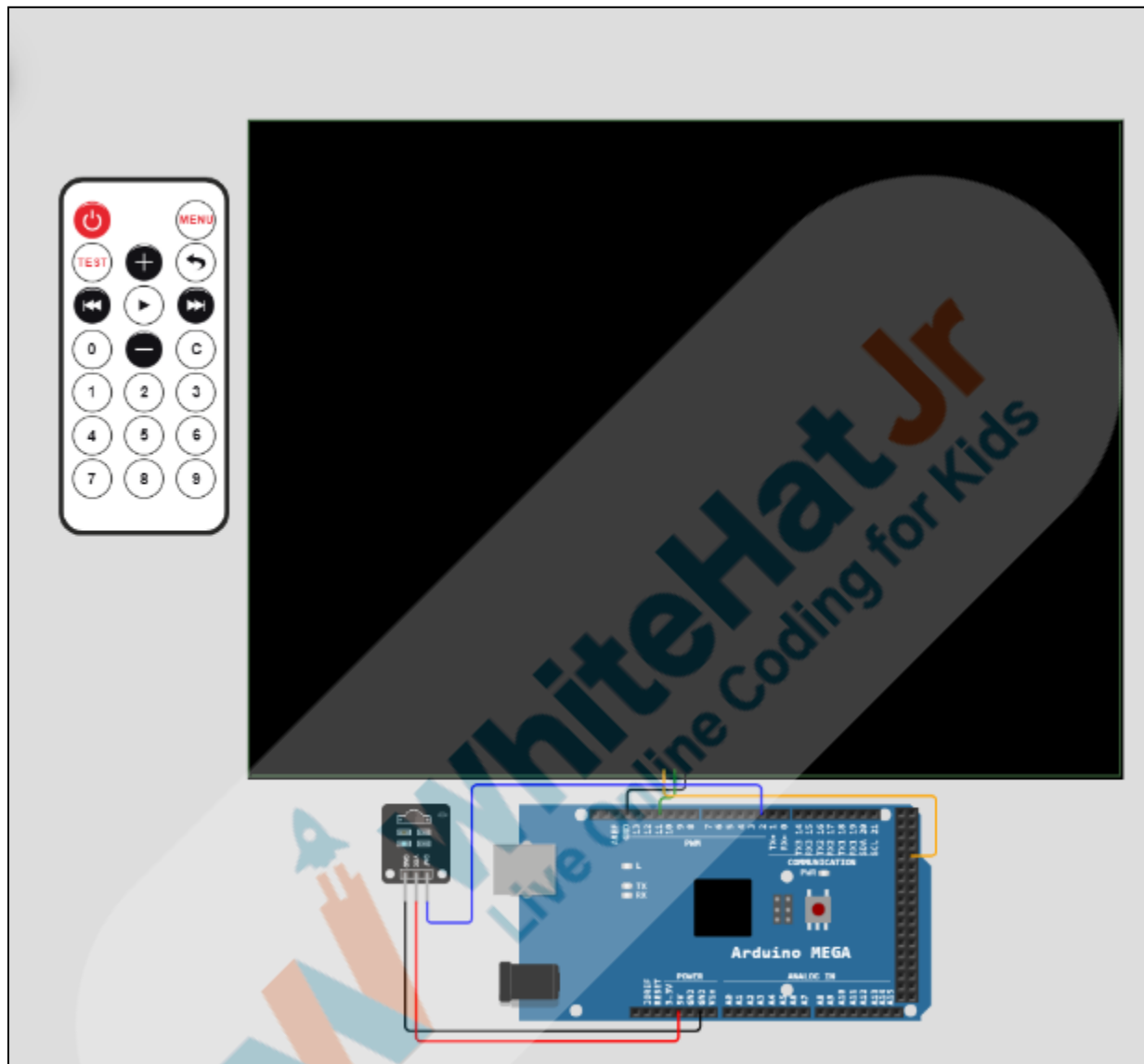
PIN connections: UNO Vs Mega

PAL TV	Arduino Uno PIN	Arduino Mega PIN
GND	GND	GND1
VIDEO	7	D29
SYNC	9	11

Do you remember the VIDEO and SYNC pins of the UNO

ESR: VIDEO pin was 7 and

<p>board?</p> <p>Yes, you are right. As you can see in the circuit, for the Mega board, the VIDEO pin is D29 and the SYNC pin is 11.</p> <p><i>Note: Output will be displayed only if VIDEO is connected to D29.</i></p> <p>Just like I mentioned in the last class, the power supply to the PALTV is 220V and is supplied automatically in the simulation. So we need not connect any pins of the Mega board to power up the TV.</p> <p>Now that the circuit is complete, let us include the IRremote.h in sketch.ino once again and test whether we get the same output as expected</p>	<p>SYNC pin was 9.</p> <p>ESR: Yes</p>
<p>CIRCUIT :</p>	



CODE:

© 2021 - WhiteHat Education Technology Private Limited.

Note: This document is the original copyright of WhiteHat Education Technology Private Limited.

Please don't share, download or copy this file without permission.

```
#include "o.h"  
#include "m.h"  
#include "e.h"  
#include "disturbance1.h"  
#include "disturbance2.h"  
#include "disturbance3.h"  
#include "font6x8.h"  
#include <IRremote.h>
```

OUTPUT:



Perfect, our TV seems to be working fine using the Arduino Mega board.

What shall we do next?

Exactly, let us start controlling the TV using the remote control.

ESR: Control the TV using the remote.

Let us understand the flow of the programme before we start coding

- The TV could be switched on or switched off. This is controlled by the power button.
- If the TV is on, we will disable the remote control receiver or else we will enable it.
- If the TV is switched on, there is a condition to check:
 1. Whether the TV was just switched on from the off state - in that case show the starting animation or disturbance images
 2. Whether the TV is already in the channel selection mode - in that case show the appropriate channel based on which button is pressed.

Now let's start coding, are you ready?

ESR: Yes!

- Let us start by creating an object of the receiver class.

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

TVout tv;

IRrecv receiver(2);
```

- We have to make a few changes to the setup() method so that the TV would switch on only through remote control. We will comment out the portion that calls **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();  
  
}
```

- 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  
}
```

- 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();  
}
```

- 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.
 - When the power button on the remote is pressed we will toggle the value of

```
TVout tv;  
IRrecv receiver(2);  
  
int tv_state = 0, last_tv_state = 0; // initially off
```

- The first button that we always press when we have the remote in hand is the power button.
- Code 162 stands for the power button on the remote.



- As you know, 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.

CODE:

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

OUTPUT:



Teacher Stops Screen Share

What do you think should be written inside the **television()** method?

ESR: The code to display the channel selection and display the programmes in the selected channel.

Do you think you will be able to do it?

ESR: Yes, I will try.

Let's try. I will guide you through it.

STUDENT-LED ACTIVITY- 15 mins	
<ul style="list-style-type: none"> Ask the student to press the ESC key to come back to the panel. Guide the student to start Screen Share. The teacher gets into Full Screen. 	
Student Initiates Screen Share	
<p style="text-align: center;"><u>ACTIVITY</u></p> <ul style="list-style-type: none"> Define the method television(). Show the starting animation only when the TV is just switched on. Switch channels when 0 or 1 is pressed. Clear screen when TV is switched off. 	
Teacher Action	Student Action
<i>The teacher guides the student to download boilerPlate code from Student activity 2.</i>	Student opens wokwi simulator and also downloads Student Activity2 .
<i>Guide the students to delete the existing files in the editor. Unzip the downloaded file and upload it in the wokwi project folder.</i>	

We have successfully connected the remote control, the IR sensor and the PAL TV to the Arduino Mega board.

Now it's time to display the two channels on the television based on the button pressed on the remote control.

Button 0 : News channel

Button 1: Dance channel

So let us write the code to display the channel selection on the TV screen.

Which is the function used to display text messages on the screen? **ESR: tv.println()**

That's correct, let us use tv.println() to display the choices of channels "News" and "Dance".

tv.println(30, 40, "SELECT A CHANNEL"); will display the text "SELECT A CHANNEL" at x position 30 and y position 40.

```
void select_channel(){  
    tv.println(30,40, "SELECT A CHANNEL");  
    tv.println(45,50,"0 : NEWS");  
    tv.println(45,60,"1 : DANCE");  
    tv.delay(100);  
}
```

Can you look at the code given and check whether there are any new methods defined?

Can you read the code and tell me what is written in each method?

That's wonderful. It's good to see you read and understand code so easily.

ESR: **news_channel()** and **dance_channel()** are two new methods defined.

ESR:
news_channel() method calls the **speaking_animation()** that displays the speaking faces and then calls the method **select_channel()** that displays the channel selection on the TV.

dance_channel() method displays the bitmaps of dancing images and then calls the method **select_channel()** that displays the channel selection on the TV

```
void news_channel(){
    speaking_animation(30,0);
    select_channel();
}

void dance_channel(int x, int y){
    for (int i = 0; i < 20; i++){
        int frame = random(0,5);
        if (frame == 0)tv.bitmap(x,y,dance1);
        else if (frame == 1)tv.bitmap(x,y,dance2);
        else if (frame == 2)tv.bitmap(x,y,dance3);
        else if (frame == 3)tv.bitmap(x,y,dance4);

        tv.delay_frame(5);
    }
    tv.clear_screen();
    tv.delay(100);
    select_channel();
}
```

Now let us define the method called **television()**.

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

Let us understand the flow of the programme before we start coding

- 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()**

Now let's start coding

1. Define the method **television()** and call the disable_receiver () method

```
void television(int button_pressed){  
    disable_receiver();  
}
```

2. Write an if condition to check the variable **tv_state**

```
void television(int button_pressed){  
    disable_receiver();  
    if (tv_state){ // tv is on  
    }  
}
```

3. Write an if condition to check the value of the variable **last_tv_state**.


```
void television(int button_pressed){
    disable_receiver();
    if (tv_state){ // tv is on
        if (last_tv_state == 0){ // tv is just turned on
        }
    }
}
```

4. If the value of the variable **last_tv_state** is zero, then call the method **starting_animation()**, call the method **channel selection()** and change the value of variable **last_tv_state** to **tv_state**.

```
void television(int button_pressed){
    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;
        }
    }
}
```

5. Write the else If part to check the value of the variable. If the value is equal to one, then write an if condition that calls **news_channel()** if button_pressed is 104 (104 is button 0 of the remote) and call **dance_channel()** method if button_pressed is 40, (40 is button 1 of the remote)

```
void television(int button_pressed){
    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 (button_pressed == 104)news_channel();
            else if (button_pressed == 48)dance_channel();
        }
    }
}
```

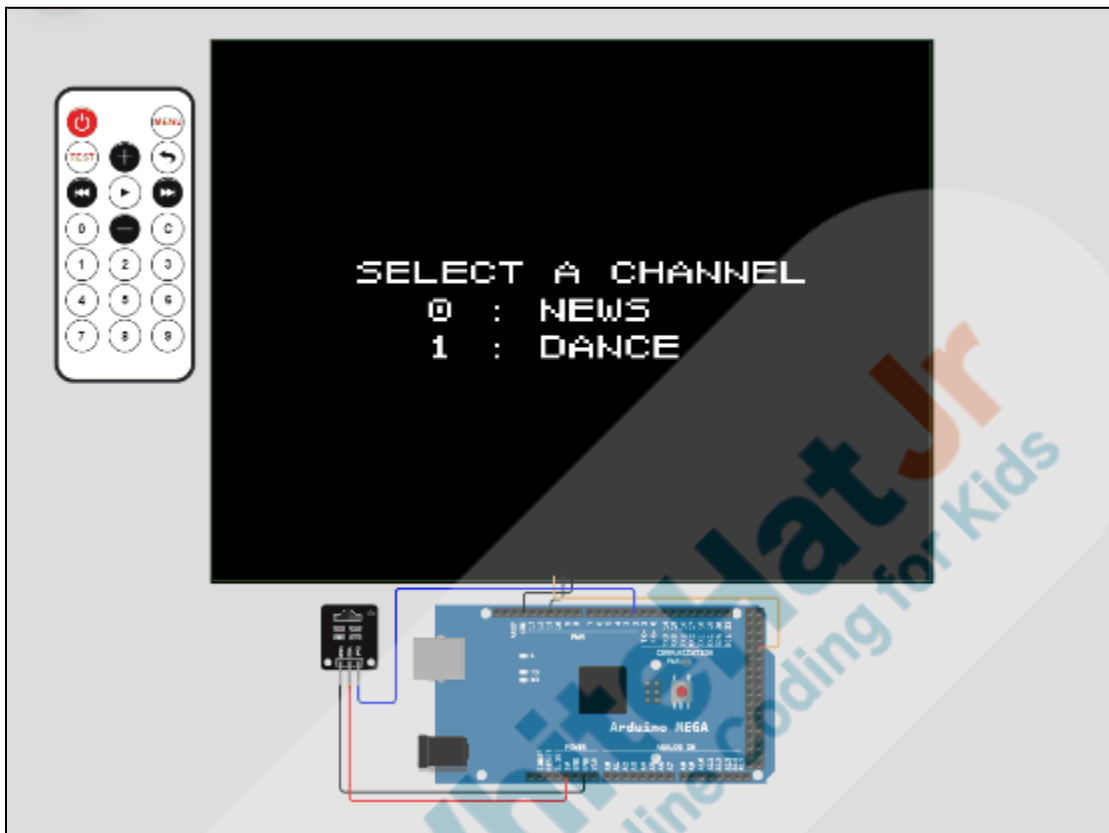
6. Write an else block for If the state of the tv is off. Inside the else block, call **tv.clear_screen()**, add a delay of 100ms and assign the value of **tv_state** to **last_tv_state**.

```
void television(int button_pressed){
    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 (button_pressed == 104)news_channel();
            else if (button_pressed == 48)dance_channel();
        }
    }
    else{ // tv is off
        tv.clear_screen();
        tv.delay(100);
        last_tv_state = tv_state;
    }
    enable_receiver();
}
```

7. Finally at the end of the method, enable the receiver one again.

```
void television(int button_pressed){  
    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 (button_pressed == 104)news_channel();  
            else if (button_pressed == 48)dance_channel();  
        }  
    }  
    else{ // tv is off  
        tv.clear_screen();  
        tv.delay(100);  
        last_tv_state = tv_state;  
    }  
    enable_receiver();  
}
```

OUTPUT:



[Click here](#) for the output reference video.

That's wonderful. You have done an amazing job today.

How do you feel?

ESR: Varied.

Teacher Guides Student to Stop Screen Share




WRAP-UP SESSION - 05 mins

Activity details

Following are the WRAP-UP session deliverables:

- Appreciate the student.
- Revise the current class activities.
- Discuss the quizzes.

WRAP-UP QUIZ

Click on In-Class Quiz	
Activity Details Following are the session deliverables: <ul style="list-style-type: none"> • Explain the facts and trivia • Next class challenge • Project for the day • Additional Activity (Optional) 	
<p style="text-align: center;"><u>FEEDBACK</u></p> <ul style="list-style-type: none"> • Appreciate and compliment the student for trying to learn a difficult concept. • Get to know how they are feeling after the session. • Review and check their understanding. 	
Teacher Action	Student Action
<p>You get “hats-off” for your excellent work!</p> <p>In the next class, we will learn about robots. We will use a tool called Webots to build a Robot.</p>	<p><i>Make sure you have given at least 2 hats-off during the class for:</i></p> <div> <div>Creatively Solved Activities  +10</div> <div>Great Question  +10</div> <div>Strong Concentration  +10</div> </div>
PROJECT OVERVIEW DISCUSSION Refer the document below in Activity Links Sections	
Teacher Clicks	<div>✕ End Class</div>
ADDITIONAL ACTIVITIES (Optional)	
Additional Activities	

ACTIVITY LINKS		
Activity Name	Description	Links
Teacher Activity 1	Simulator	https://wokwi.com/
Teacher Reference 1	PAL documentation	https://docs.wokwi.com/parts/wokwi-ir-receiver
Teacher Reference 2	TA1 Boilerplate Code	https://github.com/procodingclass/P-RO-C277-Teacher-Boilerplate-1
Teacher Reference 3	TA2 Boilerplate Code	https://github.com/procodingclass/P-RO-C277-Teacher-Boilerplate-2
Teacher Reference 4	Reference Code	https://github.com/procodingclass/P-RO-C277-Reference-Code
Teacher Reference 5	Project	https://s3-whjr-curriculum-uploads.whjr.online/27ac3b6b-14e9-4c65-b4c1-b0fa95a65b3c.pdf
Teacher Reference 6	Project Solution	https://github.com/procodingclass/P-RO-C277-Project-Solution
Teacher Reference 7	In-Class Quiz	https://s3-whjr-curriculum-uploads.whjr.online/d346afef-48ad-4913-adcc-2a18ea12e4bc.pdf
Student Activity 1	Simulator	https://wokwi.com/
Student Activity 2	Student Activity Boilerplate code	https://github.com/procodingclass/P-RO-C277-Student-Boilerplate
Student Activity 3	PAL documentation	https://docs.wokwi.com/parts/wokwi-ir-receiver