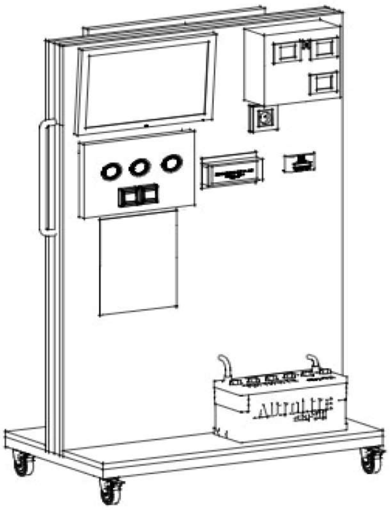
This project is funded by the European Union.

Technical guide

10 sept 2021 version

AUTONOMOUS PHOTOVOLTAIC Model



This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Index of the Document:

- description of the model
- graphics of the model (photo’s or video)
- functional scheme
- mounting instructions
- setting up instructions
- maintenance instructions
- safety instructions
- smart connection to the model

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

This project is funded by the European Union.

Description of the model

Renewable energy is a type of energy generated from renewable sources. There are many types of renewable energies, the most used by man are: solar, wind, tidal, geothermal, and aerothermal energy. These renewable sources are inexhaustible. The need for clean fuels due to the constant degradation of the atmosphere due to energy consumption in developing countries and already developed countries. They are factors that have driven the development of renewable energy.

This model will help the student to understand the basis of autonomous photovoltaic installations, it means, electrical installations that take the energy from the sun, store it and/or consume either in DC or AC modes, working always disconnected from the main electric grid.

The photovoltaic panel produces the energy that is stored in the battery. This energy is what is consumed right there. In this case, it is consumed in the form of direct current and alternating current.

The cells of a photovoltaic solar panel have silicon, and when the photons impact them, electrons are released that produce an electric current.

The regulator is responsible for transforming the output voltage of the solar panel to a suitable voltage for charging the battery. The tensions can be completely different.

The inverter is responsible for converting that direct current from the battery to alternating current like the one we have in the home plugs.

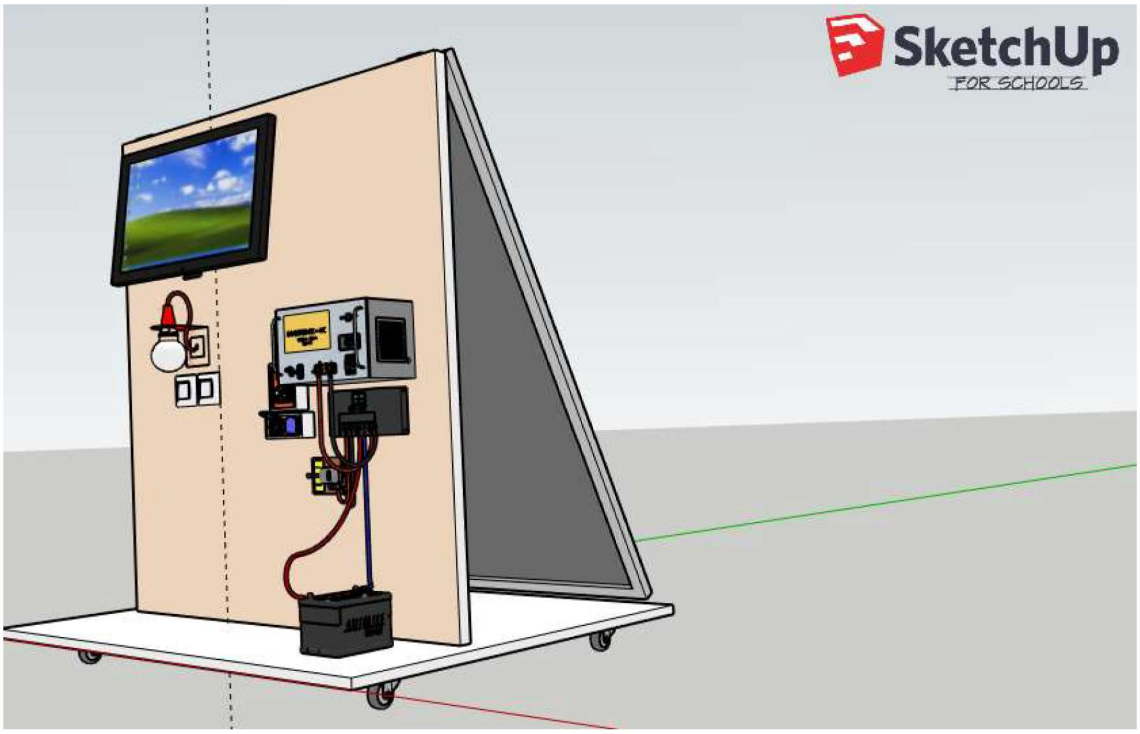
The protections are responsible of protecting the system against over currents.

This Model has been developed fulfilling the defined TECHNICAL SPECIFICATIONS for the models in the EDUTRANS PROJECT:

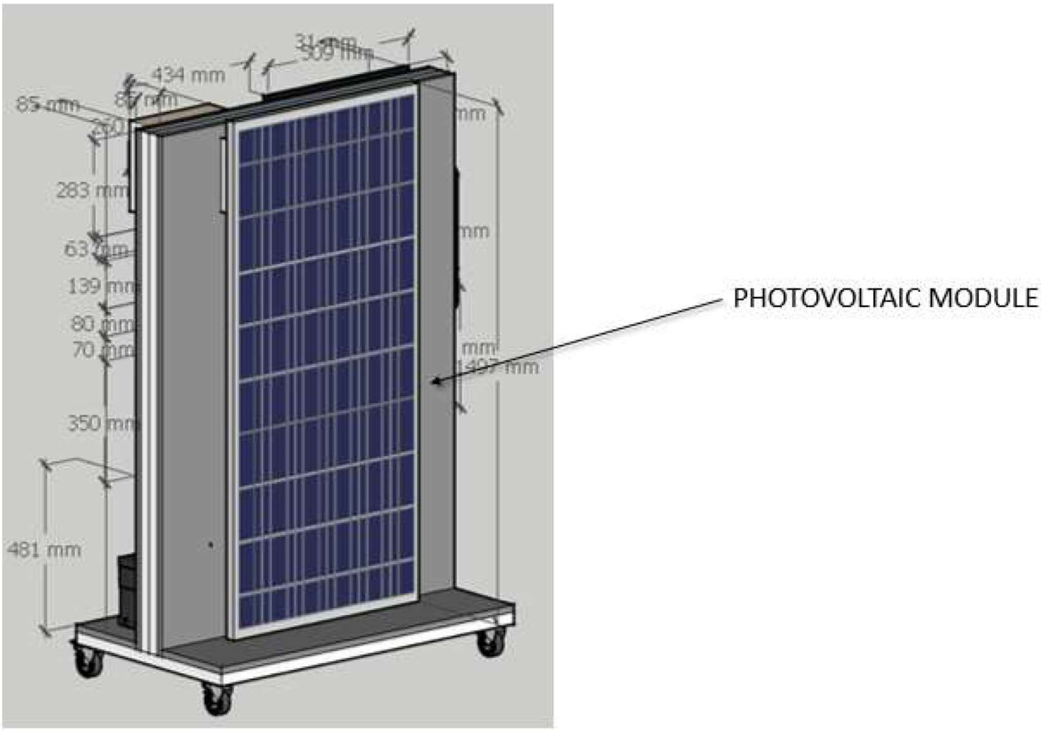
https://docs.google.com/document/d/1mJ9xmRb5Bk5AfOARahPws5V0eGGJ1OIJ/editÇ

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Graphics of the model (photo, video)

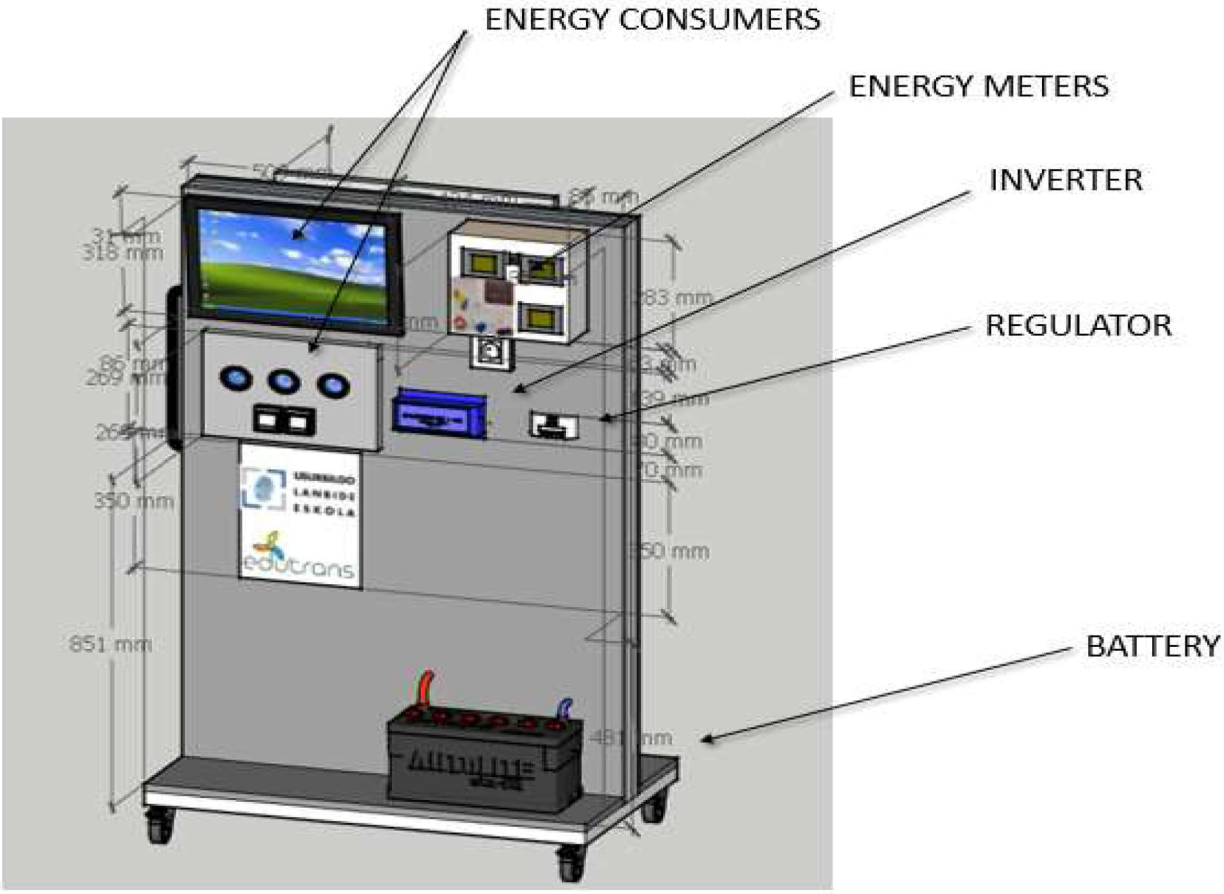


This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

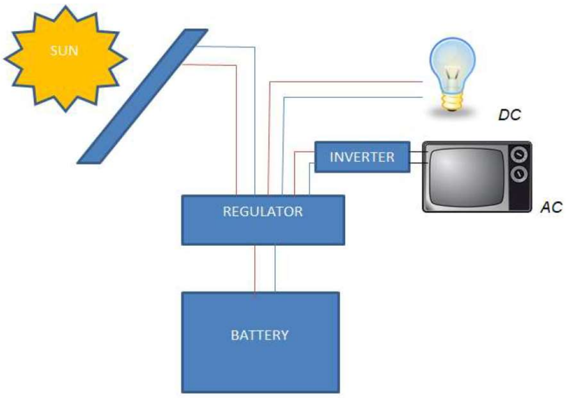
This project is funded by the European Union.

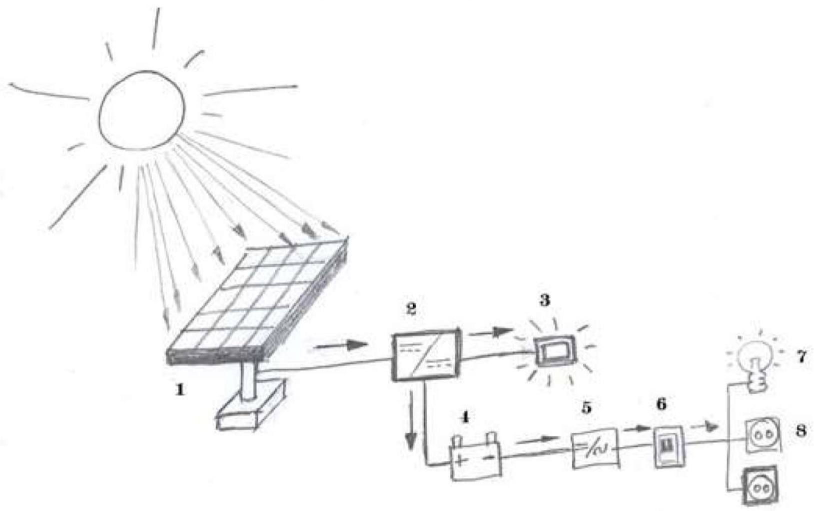


This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

This project is funded by the European Union.

Functional Scheme





1. PANEL
2. REGULATOR
3. LED LIGHT (12V)
4. BATTERY
5. INVERTER
6. PROTECTIONS
7. LIGHT (220V)
8. PLUG (220V)

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

This project is funded by the European Union.

Mounting instructions

The mounting of the elements is simple, but instructions must be read carefully.

A basic part list, composed by 2 drawings and a list of commercial elements has been developed:

PART LIST

Model: M1	M1
Desctiption:	Photovoltaic Autonomous
School:	CIFP USURBIL LHII

NUMBER	NAME	DRAWING NAME	QTY	AUTHOR	COMMERCIAL REF.	DESCRIPTION
M1-AP000	Assembly		1	Usurbil		
M1-AP001	Base	Edutrans M1 PVAut base v1	1	Usurbil	-	The movable element where we fix all the parts of the model
M1-AP002	Regulator	-	1	Commercial	VICTRON SCC 010010010	Element for regulating electrical generation, consumption and storage
M1-AP003	Inverter	-	1	Commercial	VICTRON BLUESOLAR 12V 350 VA	DC-AC converter
M1-AP004	Battery	-	1	Commercial	C100 250 Ah 12V	Energy storage
M1-AP005	Led light	-	1	Commercial		Energy consumption - DC
M1-AP006	TV	-	1	Commercial		Energy consumption - AC
M1-AP007	Electrical energy consumer	-	1	Commercial		Energy consumption - AC
M1-AP008	Energy meter	-	3	Commercial		Voltage, Intensity, Power meter
M1-AP009	PV pannel	-	1	Commercial		Energy generation from the sun - DC
M1-AP010	Fixation elements			Commercial		
M1-AP011	Cable and connectors			Commercial		
M1-AP012	Switchs		3	Commercial		on / off consumers

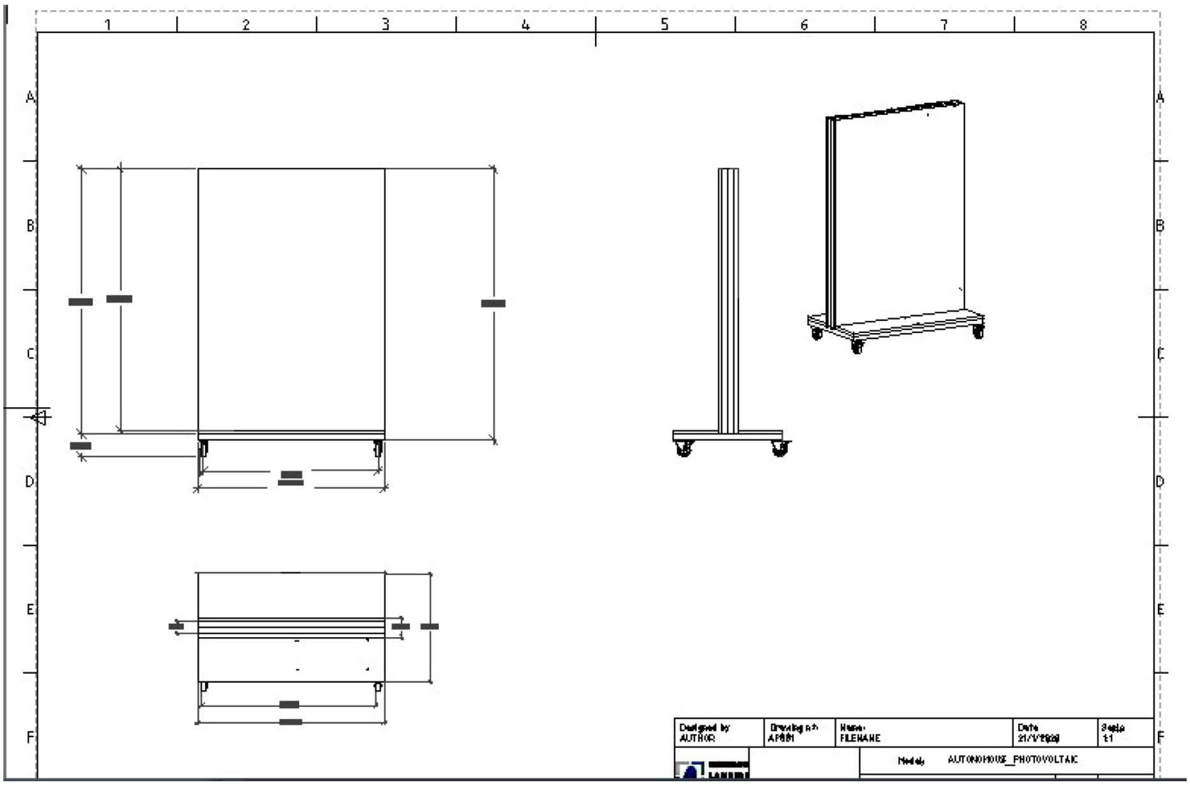
The estimated cost of all elements for Spain is of 700 Euros. It will vary depending on the country and the final elements bought.

A 3d digital model, 2D drawings and the pdf with the description of the commercial elements are included on the project.

The 3d digital model will help to understand the composition of elements and final lay-out. It can be seen in a tablet, computer or Virtual Reality glasses, and will be very helpful in the mounting planning phase.

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

A frame with wheels is needed in order to mount all the elements over it. The basic dimensions of the frame that is recommended have been indicated in the next drawing:

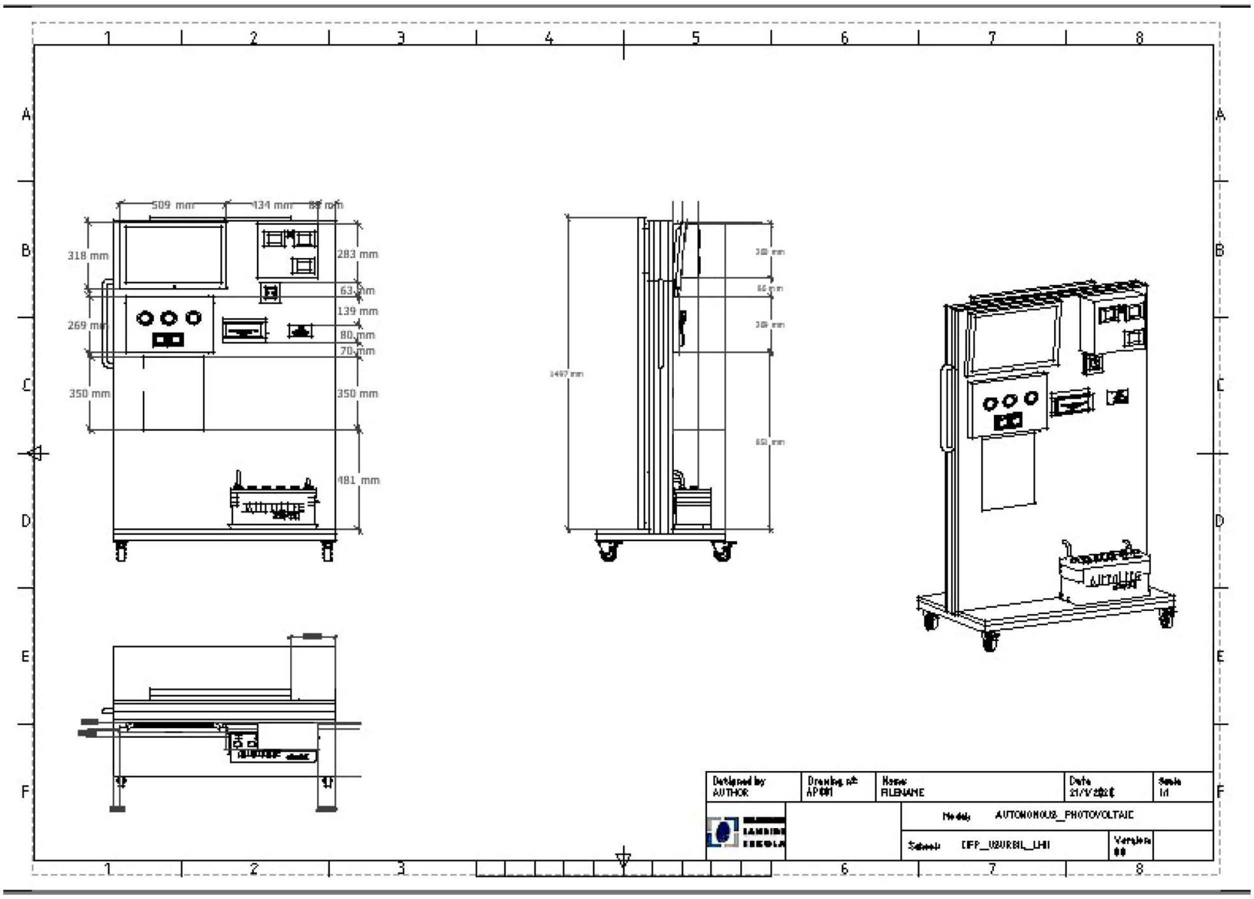


Once the frame has been mounted, the commercial elements will be placed over it.

The 3D model can be used for a proper lay-out composition.

This project is funded by the European Union.

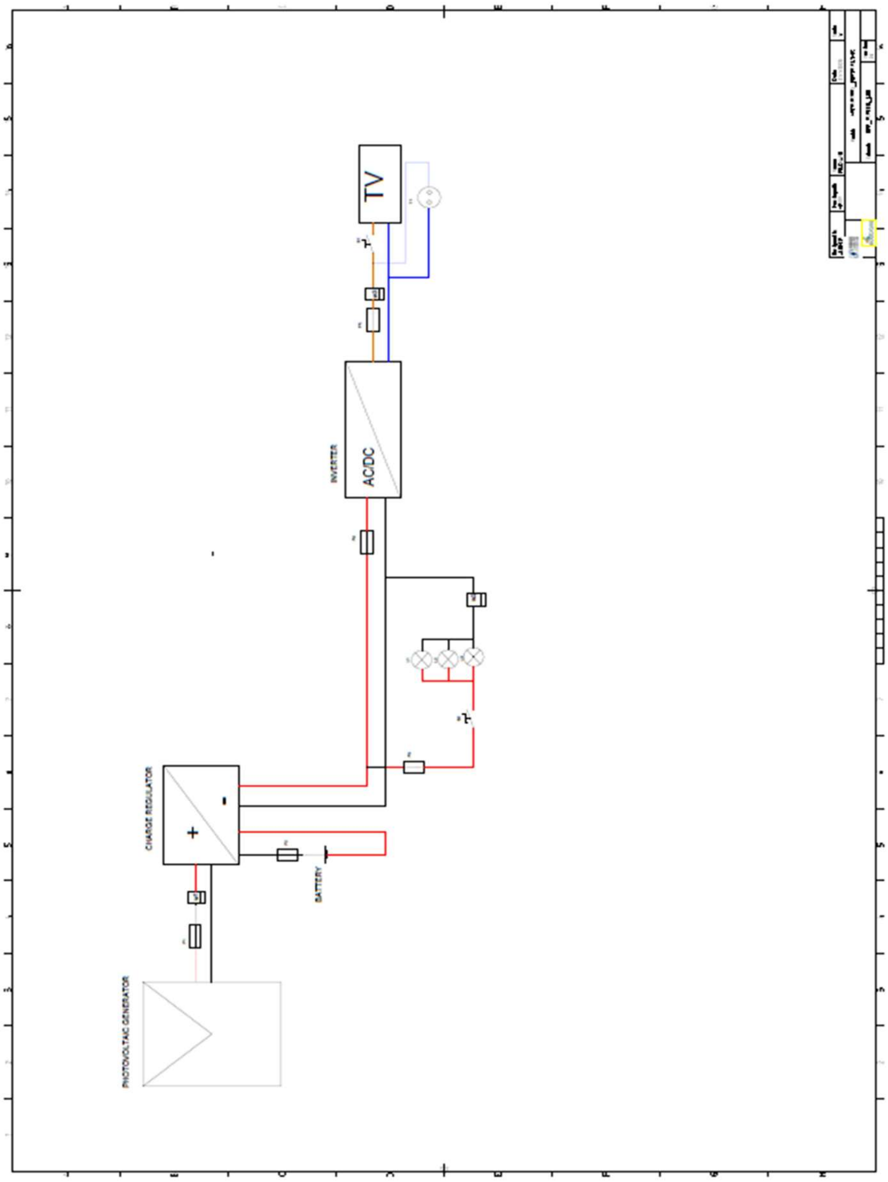
Moreover, the proposed definitive dimensions have been indicated in the assembly drawing:



Although these dimensions are orientative, it’s recommended to mount the elements in accordance to them in order to guarantee a proper function of the Model. Don’t forget to fix the photovoltaic panel at the back of the frame.

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Once the elements have been fixed, the electrical connections will be carried out according to the next scheme:



This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Setting up instructions

In order to set up the model, follow the instructions:

1. Switch on the protections that are placed inside the box.
2. Put the model in the exterior, with the solar panel facing the sun.
3. Measure the voltage value in the battery. It must be over 24V.
4. In order to be sure that the panel is working, with a tester measure the voltage in the photovoltaic panel's bornes.
5. Observe the measurers on the box, the voltage value measured with the tester must be read on the corresponding measurer of the panel.
6. Switch on the lights, observe that they work and read DC current and voltage values in the corresponding measurer.
7. Connect the TV and observe the AC current and voltage values in the corresponding measurer.
8. Once this basic setting up instructions have been fulfilled, the model is ready for carrying out the tests described in the STUDENT'S HANDBOOK.

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Maintenance instructions

The critical element for maintenance in the model is the battery. Therefore, all described instructions that appear in the battery prospect must be fulfilled carefully.

The model must be stored in the interior, and the protections must be disconnected every time we are not using it for teaching purposes.

The brake of the wheels must be activated every time we are not transporting the model.

The model must be transported carefully inside school's installations, and it is recommended to protect it with specific material every time that we want to move it to other installations that are outside the school.

Protect specifically the TV and measurer's screens every time we are not using the model.

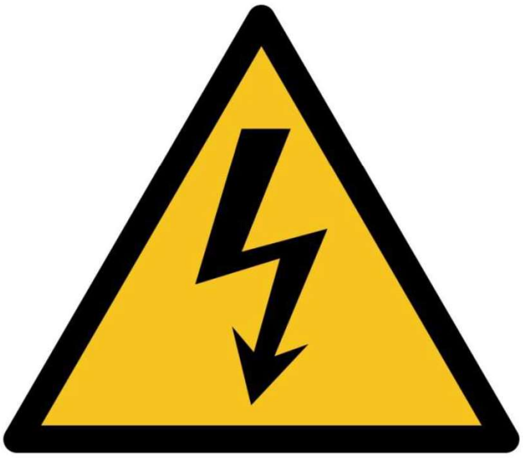
This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

This project is funded by the European Union.

Safety instructions

Danger, electric shock risk!

The model has parts that work at 230V, always manipulate it using electricity purpose homologated gloves.



Cover the battery bornes with proper material, avoiding direct contact for people with them.

Other purposes that are not the specified ones on the TEACHER HANDBOOK are not recommended.

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

This project is funded by the European Union.

SMART connection to the model

This is an extra feature we developed for Model 2, but can be implemented on Model 1 also. With that, basic parameters of the model (generated and consumed current, tension and power) can be monitored remotely using an internet connection. This has been developed by Zubigune's Ander Mauleon for Usurbilgo Lanbide Eskola, and represents a cheap monitoring solution using Arduino and Raspberry Pi Open Source technology.

Results can be seen in this link:

https://edutrans.lekuiot.com/fotoredsens.html

By now, the instructions are only in Spanish:

Material

Para el desarrollo de la parte de monitorización se han utilizado unos sensores de medición de tensión y corriente, estos mismos dan la información del consumo en W y kWh, mostrando el factor de potencia.

Los sensores son unos PZEM-004T-100A, cuyas características del Datasheet

https://innovatorsguru.com/wp-content/uploads/2019/06/PZEM-004T-V3.0-Datasheet-User-Manual.pdf

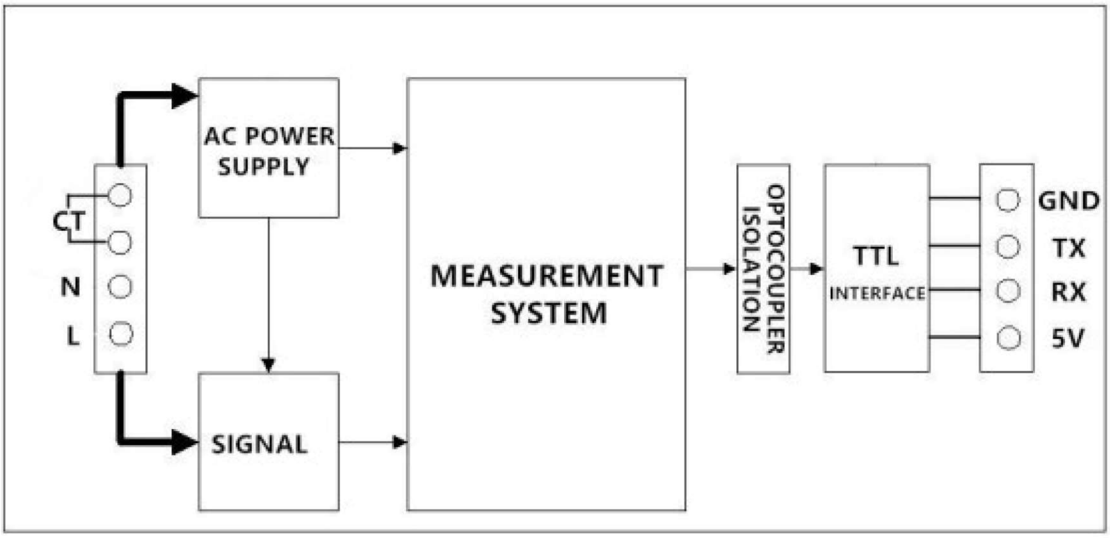
son las siguientes:

Voltage:
Rango de medida:80～260V, Resolución: 0.1V
Corriente:
Rango de medida: 0～100A. Medida inicial: 0.02A(PZEM-004T-100A), Resolución: 0.001A
Potencia activa:
Rango de medida: 0～23kW. Medida inicial 0.4W, Resolución: 0.1W

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

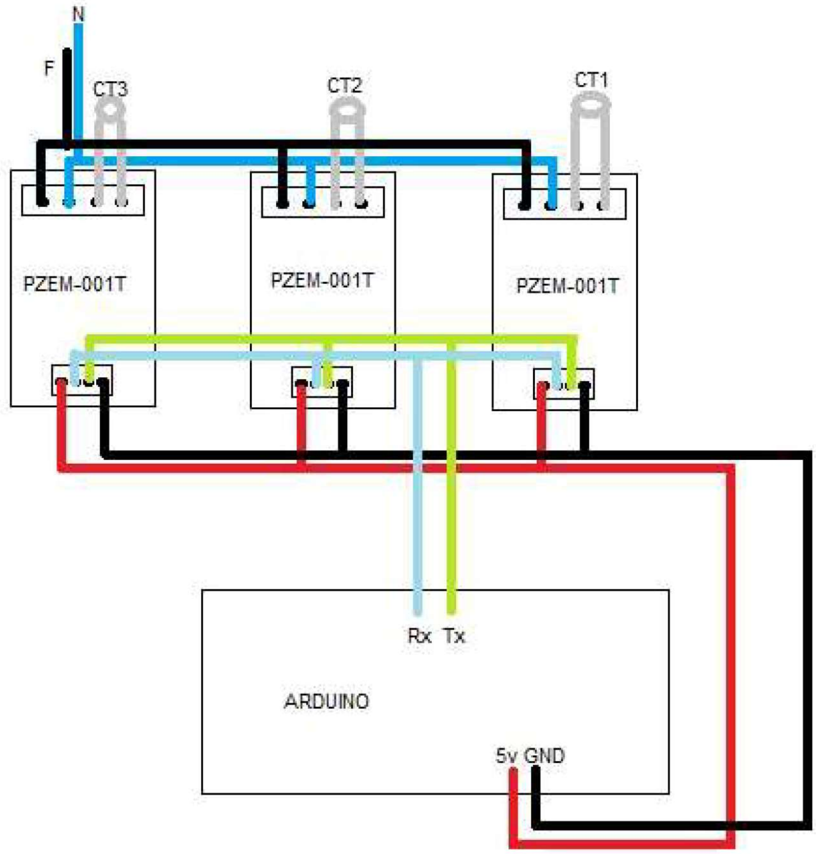
Factor de potencia:
Rango de medida: 0.00～1.00. Resolución 0.01
Frecuencia:
Rango de medida: 45Hz～65Hz. Resolución: 0.1Hz
Energía (Activa):
Rango de medida: 0～9999.99kWh, Resolución: 1Wh
＜10kWh, la unidad a mostrar es Wh(1kWh=1000Wh), hasta: 9999Wh
≥10kWh, la unidad a mostrar es kWh, hasta: 9999.99kWh
Reset de las medidas: por software.

El esquema de las conexiones y partes de las que se compone son las siguientes:



Esquema de conexion

El esquema de conexión es el siguiente, los CTs son las bobinas que rodean al cable ahí donde queremos medir la corriente, la F y N son la fase y el neutro ahí donde queremos medir la diferencia de potencial, la alimentación de 5v y GND se realiza a través de los pines de Arduino, y los pines de Rx y Tx se usan para pasar los datos de las mediciones a Arduino que posteriormente se pasaran por puerto serie a la Raspberry pi.



Programa de Arduino uno.

El programa de Arduino realizado con el IDE de Arduino, para leer los datos y sacarlos por el puerto serie sería el siguiente.

```
#include <PZEM004Tv30.h>
#define TX
#define RX 12
```


This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
PZEM004Tv30 pzem1(11, 12, 0x01);
PZEM004Tv30 pzem2(11, 12, 0x02);
PZEM004Tv30 pzem3(11, 12, 0x03);


void setup() {
 Serial.begin(115200);
}


void loop() {
 Serial.println("pzem1====================");
 float voltage = pzem1.voltage();
 if (voltage != NAN) {
  Serial.print("Voltage: "); Serial.print(voltage); Serial.println("V");
 } else {
  Serial.println("Error reading voltage");
 }


 float current = pzem1.current();
 if (current != NAN) {
  Serial.print("Current: "); Serial.print(current); Serial.println("A");
 } else {
  Serial.println("Error reading current");
 }


 float power = pzem1.power();
 if (current != NAN) {
  Serial.print("Power: "); Serial.print(power); Serial.println("W");
 } else {
```

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
  Serial.println("Error reading power");
 }

 float energy = pzem1.energy();
 if (current != NAN) {
  Serial.print("Energy: "); Serial.print(energy, 3); Serial.println("kWh");
 } else {
  Serial.println("Error reading energy");
 }

 float frequency = pzem1.frequency();
 if (current != NAN) {
  Serial.print("Frequency: "); Serial.print(frequency, 1); Serial.println("Hz");
 } else {
  Serial.println("Error reading frequency");
 }

 float pf = pzem1.pf();
 if (current != NAN) {
  Serial.print("PF: "); Serial.println(pf);
 } else {
  Serial.println("Error reading power factor");
 }


 Serial.println();
 Serial.println("pzem2=====================");
```


This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
voltage = pzem2.voltage();
if (voltage != NAN) {
  Serial.print("Voltage: "); Serial.print(voltage); Serial.println("V");
} else {
  Serial.println("Error reading voltage");
}

current = pzem2.current();
if (current != NAN) {
  Serial.print("Current: "); Serial.print(current); Serial.println("A");
} else {
  Serial.println("Error reading current");
}

power = pzem2.power();
if (current != NAN) {
  Serial.print("Power: "); Serial.print(power); Serial.println("W");
} else {
  Serial.println("Error reading power");
}

energy = pzem2.energy();
if (current != NAN) {
  Serial.print("Energy: "); Serial.print(energy, 3); Serial.println("kWh");
} else {
  Serial.println("Error reading energy");
}
```

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
frequency = pzem2.frequency();
if (current != NAN) {
  Serial.print("Frequency: "); Serial.print(frequency, 1); Serial.println("Hz");
} else {
  Serial.println("Error reading frequency");
}

pf = pzem2.pf();
if (current != NAN) {
  Serial.print("PF: "); Serial.println(pf);
} else {
  Serial.println("Error reading power factor");
}

Serial.println();
Serial.println("pzem3=====================");

voltage = pzem3.voltage();
if (voltage != NAN) {
  Serial.print("Voltage: "); Serial.print(voltage); Serial.println("V");
} else {
  Serial.println("Error reading voltage");
}

current = pzem3.current();
if (current != NAN) {
  Serial.print("Current: "); Serial.print(current); Serial.println("A");
} else {
```


This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
  Serial.println("Error reading current");
}

power = pzem3.power();
if (current != NAN) {
  Serial.print("Power: "); Serial.print(power); Serial.println("W");
} else {
  Serial.println("Error reading power");
}

energy = pzem3.energy();
if (current != NAN) {
  Serial.print("Energy: "); Serial.print(energy, 3); Serial.println("kWh");
} else {
  Serial.println("Error reading energy");
}

frequency = pzem3.frequency();
if (current != NAN) {
  Serial.print("Frequency: "); Serial.print(frequency, 1); Serial.println("Hz");
} else {
  Serial.println("Error reading frequency");
}

pf = pzem3.pf();
if (current != NAN) {
  Serial.print("PF: "); Serial.println(pf);
} else {
```

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
    Serial.println("Error reading power factor");
  }

  Serial.println();

  Serial.println();
  delay(2000);
}
```

Raspberry pi

En la parte de monitorización se ha usado una Raspberry pi, para poder monitorizar en pantalla los valores, a la vez que se muestran unas diapositivas que muestran unas breves explicaciones del funcionamiento.

A su vez esta envía los datos a otra Raspberry pi, que hace de servidor Web. En esta segunda se aloja la pagina web donde se puede consultar el funcionamiento y ver los datos en tiempo real con un pequeño lag.

Conexionado

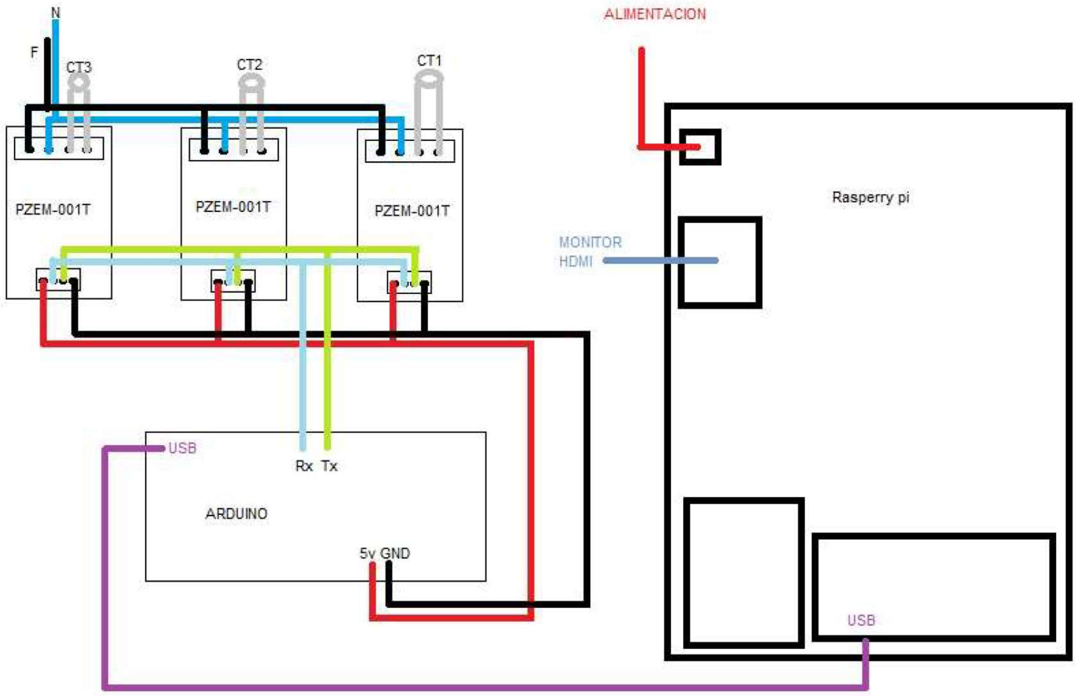
Se muestra como se conecta el Arduino a la Raspberry pi mediante un cable USB, mediante el cual se alimenta Arduino, y los módulos PZEM, a través de éste también se comunica mediante comunicación serie para enviar los datos de los sensores.

La Raspberry pi a su vez necesita una fuente de alimentación y un cable HDMI para conectarse con el monitor.

La transferencia de datos de la Raspberry pi de expositor con la Raspberry pi que hace se servidor web es mediante wifi, o bien compartiendo con el móvil o con un router 4g.

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Esquema:



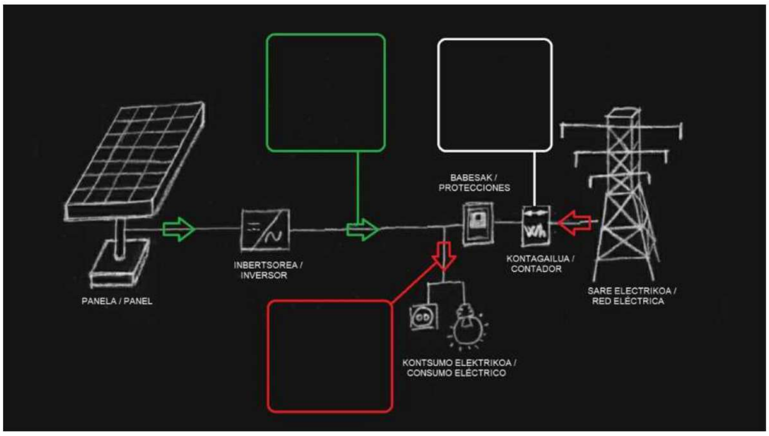
Configuración y programación Raspberry pi.

Para la demostración en pantalla de la raspberry pi nada mas arrancar , se configura para que inicie automáticamente con un usuario. Nada mas arrancar se utiliza un script que inicia el visor de imágenes, en este caso se usa el potente visor de imágenes feh, este muestra las diapositivas a modo de carrusel y en una de ellas aparecen los valores de las mediciones. Esta diapositiva que aparecen los valores se monta mediante Pillow, es una biblioteca adicional gratuita y de código abierto para el lenguaje de programación Python que agrega soporte para abrir, manipular y guardar muchos formatos de archivo de imagen diferentes.

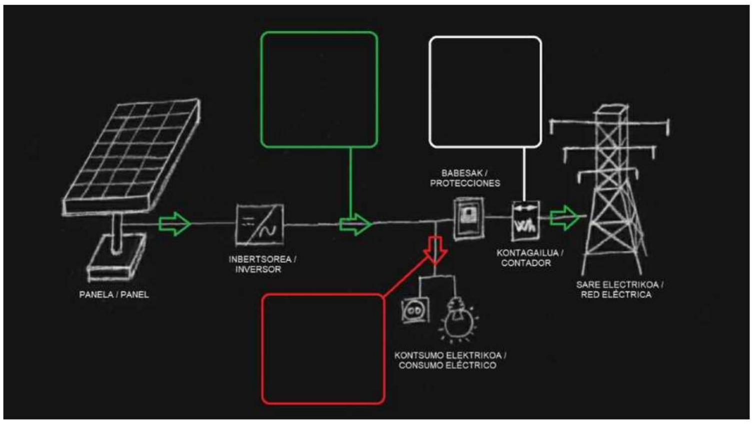
This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Las imágenes base sobre las que trabaja Pillow para introducir los datos son consumiendo, generando y sin consumo, para poder ver las imágenes de las flechas de la dirección de la corriente.

Consumiendo



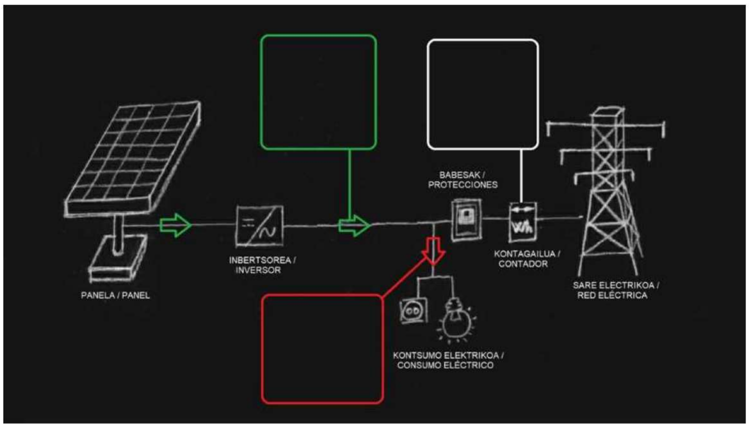
Generando



This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

This project is funded by the European Union.

Sin consumo



Recogida de datos desde puerto serie

Recogemos los datos mediante un script de Python.

```
IMPORT SHUTIL
FROM OS IMPORT REMOVE
SER = SERIAL.SERIAL(PORT = "/DEV/TTYACM0", BAUDRATE = 115200)
IF OS.PATH.EXISTS("/HOME/PI/ERRORTTY.TXT"):
   REMOVE("/HOME/PI/ERRORTTY.TXT")
TRY:
   WHILE TRUE:
      READ = SER.READLINE().STRIP()
      IF READ == "GRABAR":
         WHILE READ == "GRABAR":
            PRINT (READ)
#           PRINT ('READ = GRABAR')
            Z = OPEN('DATOS.PY', 'W')
            Z.WRITE("NAN = 0")
            READ = SER.READLINE().STRIP()
            WHILE READ != 'GRABAR':
               Z.WRITE("\N")
               Z.WRITE(READ)
               READ=SER.READLINE().STRIP()
                   PRINT(READ)
            Z.CLOSE()
```

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
            HORA=TIME.STRFTIME("%H:%M - %D/%M/%Y")
            SHUTIL.COPY('/HOME/PI/DATOS.PY','/HOME/PI/DATOSE.PY')
            Y = OPEN('DATOSE.PY', 'A')
            Y.WRITE("\N")
            Y.WRITE("HORA =")
            Y.WRITE(""")
            Y.WRITE(HORA)
            Y.WRITE(""")
            Y.CLOSE()
#                   TIME.SLEEP(1)
EXCEPT EXCEPTION AS ERROR:
    X = OPEN('/HOME/PI/ERRORTTY.TXT', 'W')
    X.WRITE(ERROR)
#EXCEPT KEYBOARDINTERRUPT:
#    PRINT "SALIENDO DEL PROGRAMA POR INTERRUPCION DE TECLADO"
FINALLY:
    SER.CLOSE()
```

Montaje de imágenes para el carrusel informativo.

Una vez tenemos estas plantillas introducimos los valores según la posición relativa en la pantalla mediante Pillow. Este es el programa usado. Programado en Python.

```
IMPORT IMPORTLIB
IMPORT SHUTIL
IMPORT OS
IMPORT DATOSE
FROM DATOSE IMPORT *
FROM PIL IMPORT IMAGEFONT
FROM PIL IMPORT IMAGE
FROM PIL IMPORT IMAGEDRAW
IMPORT TIME
DEF MEASURE_TEMP():
    TEMP=OS.POPEN("VCGENCMD MEASURE_TEMP").READLINE()
    RETURN (TEMP.REPLACE("TEMP=",""))
V=0
FONT = IMAGEFONT.TRUETYPE("/HOME/PI/DEJAVUSANS.TTF",25)
IF V < 100:
    I=0
```

```
WHILE I < 1:
    FROM DATOSE IMPORT *
    IMPORTLIB.RELOAD(DATOSE)
    HORA = TIME.STRFTIME("%H:%M    %D/%M/%Y")
    TEMP = MEASURE_TEMP()
    CONSUMONUM = (I2 - I1)
    IF CONSUMONUM < 0 :
        BOARD = IMAGE.OPEN("/HOME/PI/GENERANDO.PNG")
        I3 = -I3
    ELIF CONSUMONUM > 0 :
        BOARD = IMAGE.OPEN("/HOME/PI/CONSUMIENDO.PNG")
    ELSE:
        BOARD = IMAGE.OPEN("/HOME/PI/SINCONSUMO.PNG")
    CONSUMO = STR(CONSUMONUM)
    P3NN = (P2 - P1)
    P3N = ROUND(P3NN,2)
    KWH3NN = (KWH2 - KWH1)
    KWH3N = ROUND(KWH3NN,3)
    I1 = STR(I1)
    I2 = STR(I2)
    I3 = STR(I3)
    V1 = STR(V1)
    V2 = STR(V2)
    V3 = STR(V3)
    P1 = STR(P1)
    P2 = STR(P2)
    P3 = STR(P3N)
    KW1 = STR(KWH1)
    KW2 = STR(KWH2)
    KW3 = STR(KWH3N)
    F1 = STR(HZ1)
    F2 = STR(HZ2)
    F3 = STR(HZ3)
    PF1 = STR(FP1)
    PF2 = STR(FP2)
    PF3 = STR(FP3)
    DRAW = IMAGEDRAW.DRAW(BOARD)
    DRAW.TEXT((5, 870), HORA , FONT=FONT, FILL="WHITE")
    DRAW.TEXT((260, 840), TEMP , FONT=FONT, FILL="WHITE")
    DRAW.TEXT((5, 840), "RASPBERRYPI T°CPU" , FONT=FONT, FILL="WHITE")
    #PLACAS
    DRAW.TEXT((590,110), "                  V", FONT=FONT, FILL="GREEN")
    DRAW.TEXT((600,110), V1, FONT=FONT, FILL="GREEN")
    DRAW.TEXT((590,140), "                  A", FONT=FONT, FILL="GREEN")
    DRAW.TEXT((600,140), I1, FONT=FONT, FILL="GREEN")
```


This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

This project is funded by the European Union.

education & energy transition

```
        DRAW.TEXT((590,170), "                W", FONT=FONT, FILL="GREEN")
        DRAW.TEXT((600,170), P1, FONT=FONT, FILL="GREEN")
        DRAW.TEXT((590,200), "                HZ", FONT=FONT, FILL="GREEN")
        DRAW.TEXT((600,200), F1, FONT=FONT, FILL="GREEN")
        DRAW.TEXT((590,230), "                (PF)", FONT=FONT, FILL="GREEN")
        DRAW.TEXT((600,230), PF1, FONT=FONT, FILL="GREEN")
        DRAW.TEXT((590,260), "                KWH", FONT=FONT, FILL="GREEN")
        DRAW.TEXT((600,260), KW1, FONT=FONT, FILL="GREEN")
        #ENCHUFE
        DRAW.TEXT((595,690), "                V", FONT=FONT, FILL="RED")
        DRAW.TEXT((600,690), V2, FONT=FONT, FILL="RED")
        DRAW.TEXT((595,720), "                A", FONT=FONT, FILL="RED")
        DRAW.TEXT((600,720), I2, FONT=FONT, FILL="RED")
        DRAW.TEXT((595,750), "                W", FONT=FONT, FILL="RED")
        DRAW.TEXT((600,750), P2, FONT=FONT, FILL="RED")
        DRAW.TEXT((595,780), "                HZ", FONT=FONT, FILL="RED")
        DRAW.TEXT((600,780), F2, FONT=FONT, FILL="RED")
        DRAW.TEXT((595,810), "                (PF)", FONT=FONT, FILL="RED")
        DRAW.TEXT((600,810), PF2, FONT=FONT, FILL="RED")
        DRAW.TEXT((595,840), "                KWH", FONT=FONT, FILL="RED")
        DRAW.TEXT((600,840), KW2, FONT=FONT, FILL="RED")
        #RED
        DRAW.TEXT((970,115), "                V", FONT=FONT, FILL="WHITE")
        DRAW.TEXT((975,115), V3, FONT=FONT, FILL="WHITE")
        DRAW.TEXT((970,145), "                A", FONT=FONT, FILL="WHITE")
        DRAW.TEXT((975,145), I3, FONT=FONT, FILL="WHITE")
        DRAW.TEXT((970,175), "                W", FONT=FONT, FILL="WHITE")
        DRAW.TEXT((975,175), P3, FONT=FONT, FILL="WHITE")
        DRAW.TEXT((970,205), "                HZ", FONT=FONT, FILL="WHITE")
        DRAW.TEXT((975,205), F3, FONT=FONT, FILL="WHITE")
        DRAW.TEXT((970,235), "                KWH", FONT=FONT, FILL="WHITE")
        DRAW.TEXT((975,235), KW3, FONT=FONT, FILL="WHITE")
        #DRAW.TEXT((970,265), "                (PF)", FONT=FONT, FILL="WHITE")
        #DRAW.TEXT((975,265), PF3, FONT=FONT, FILL="WHITE")
        #GUARDAR
        #BOARD.SAVE("/HOME/PI/IMAGENES/DASHBOARD/PRUEBA.PNG")
        #SHUTIL.COPY('/HOME/PI/IMAGENES/DASHBOARD/PRUEBA.PNG','/HOME/PI/IMAGENES/DASHBOARD/CAMBIADA
.PNG')
        #IM = IMAGE.OPEN("/HOME/PI/IMAGENES/DASHBOARD/CAMBIADA.PNG")
        BOARD.SAVE("/HOME/PI/IMAGENES/CAMBIAD.PNG")
        RGB_BOARD = BOARD.CONVERT('RGB')
        RGB_BOARD.SAVE("/HOME/PI/IMAGENES/CAMBIAD.JPG")
        SHUTIL.COPY("./IMAGENES/CAMBIAD.JPG", "./IMAGENES/DASHBOARD/CAMBIADA.JPG")
        TIME.SLEEP(1)
ELSE:
```

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
PRINT("ERROR")
```

Presentación en pantalla de las imágenes.

Mediante el siguiente script en bash, se realiza la carga automática de las imágenes y presentación en pantalla.

```
#!/BIN/BASH
COUNT=1
WHILE [ $COUNT -LE 1000 ]
DO
 VIDEO=1
 WHILE [ $VIDEO -LE 1 ]
 DO
  CLEAR
  FEH --QUIET --AUTO-ZOOM --FULLSCREEN --ON-LAST-SLIDE QUIT --RELOAD 2 --SLIDESHOW-DELAY 120.0 /HOME/PI/IMAGENES/DASHBOARD/CAMBIADA.JPG
  FEH --QUIET --AUTO-ZOOM --FULLSCREEN --ON-LAST-SLIDE QUIT --RELOAD 2 --SLIDESHOW-DELAY 10.0 /HOME/PI/IMAGENES/EDUCATIVO/*
  CLEAR
  VIDEO=$(($VIDEO + 1))
 DONE
DONE
```

Para el paso de información de una Raspberry pi a la otra lo realizamos mediante el envío a través de un túnel SSH de un archivo con los datos de los sensores. En el servidor web recogemos esos datos y volvemos a utilizar Pillow para montar la imagen y mostrarla en la página web mediante código HTML.

Para el inicio de los scripts se usa un proceso demonio del sistema llamado crontab o bien se levanta un servicio específico.

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.