

1x bottom 48x14

all measurements in **centimeters** (cm)
created from plywood of 0.9

1x back 48x21

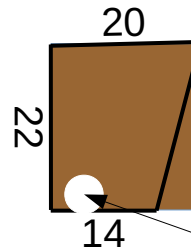
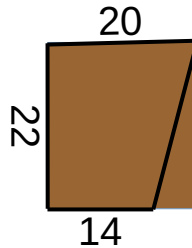
1-2x glass 48x21
(depends on your needs, type, depth...)

1x top 52*20
(overhanging on all sides)

1x digits holder 48x15

2x sides 22*20 :

cut as:
When looking into digits
left side, no hole(Arduino), right side, hole for cable for power



hole for cable
D=cca 5cm
cca 2cm from left,
removing cca 1cm from bottom

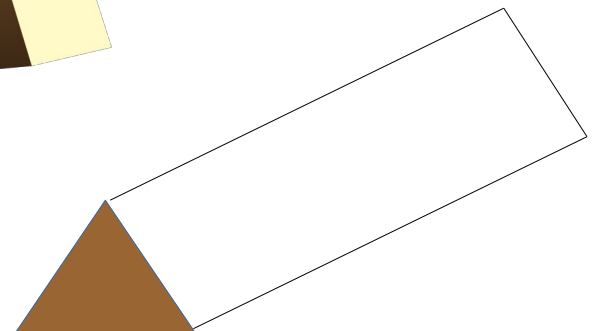
26x 4x4x2
width of 2mm
holes 4mm in diameter










major **screws** - 26*4=cca 150 of 1.2 cm screws (0.035x0.12)
to connect L peaces with wood
warning, if used without L peace, will go **through**!
You will need few more screws to:
* connect pillars and triangular blockers
* some bolts+nuts to connect roof
but those do not need to be precises

5x 1x1x21_[46] (10x scaled)
to create rails for glass
4x vertical (21)
1x horizontal (46cm)

3x 2x2xcca2.8 (triangular) x46 (10x scaled)
to create
1) top and bottom block for glass
2) to create bottom **back** for glass (the digits
do not touch ground)

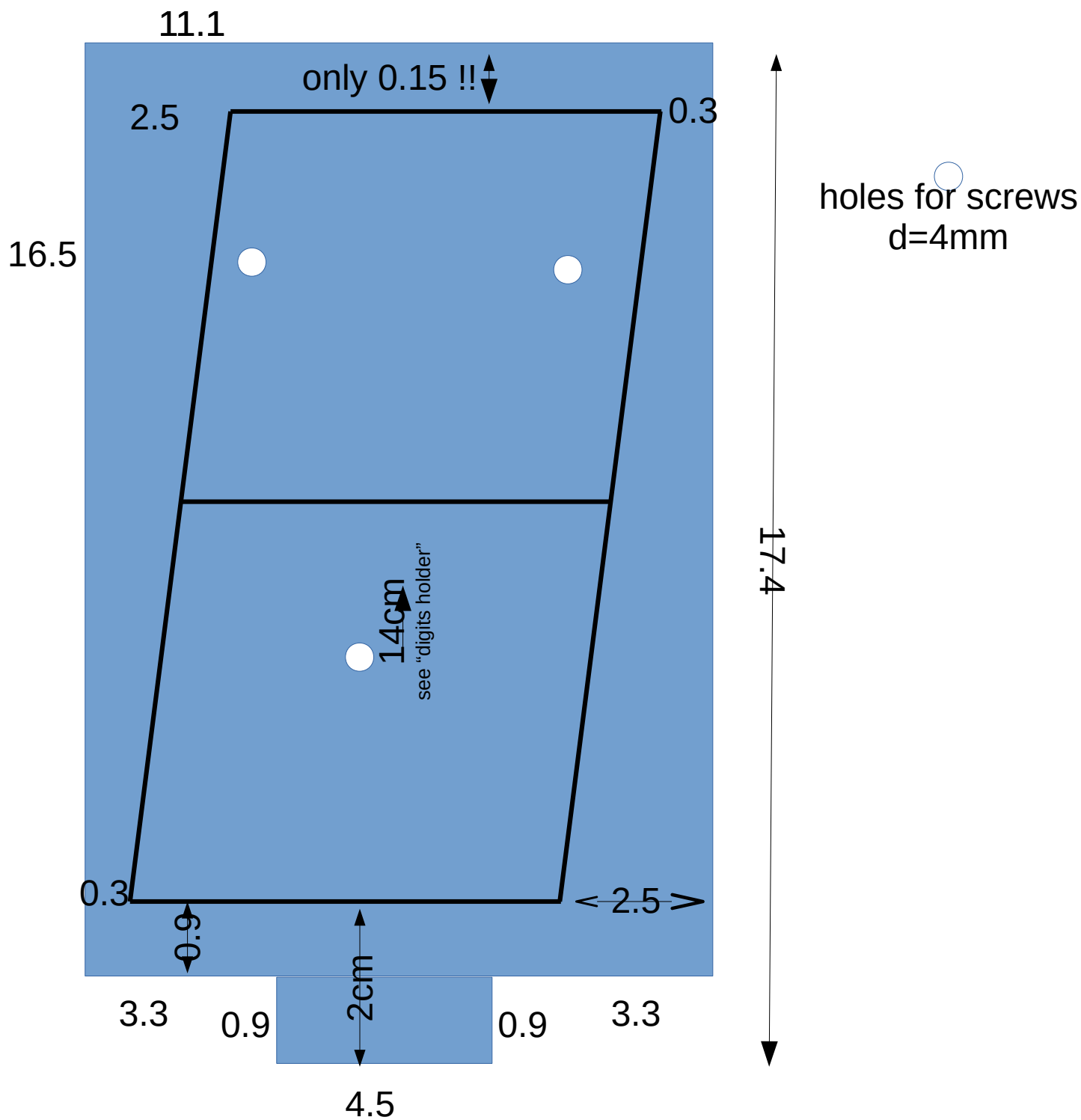


electric parts: https://www.sparkfun.com/wish_lists/148231

Product	Subtotal	Qty
 Jumper Wire - 0.1", 6-pin, 4" PRT-10366	\$3.80	4 ADD TO CART
 SparkFun Large Digit Driver WIG-13279 RoHS	\$31.80	4 ADD TO CART
 7-Segment Display - 6.5" (Red) COM-08530 RoHS	\$75.80	4 ADD TO CART
 SparkFun USB Mini-B Cable - 6 Foot CAB-11301 RoHS	\$3.95	1 ADD TO CART
 SparkFun RedBoard - Programmed with Arduino DEV-13975 RoHS	\$19.95	1 ADD TO CART
 Jumper Wires Premium 6" M/F Pack of 10 PRT-09140 RoHS	\$3.95	1 ADD TO CART
 Wall Adapter Power Supply - 12VDC 600mA TOL-09442 RoHS	\$5.95	1 ADD TO CART
Total: \$145.20		ADD ALL THIS STUFF

to build 4digits clocks based to <https://learn.sparkfun.com/tutorials/large-digit-driver-hookup-guide>

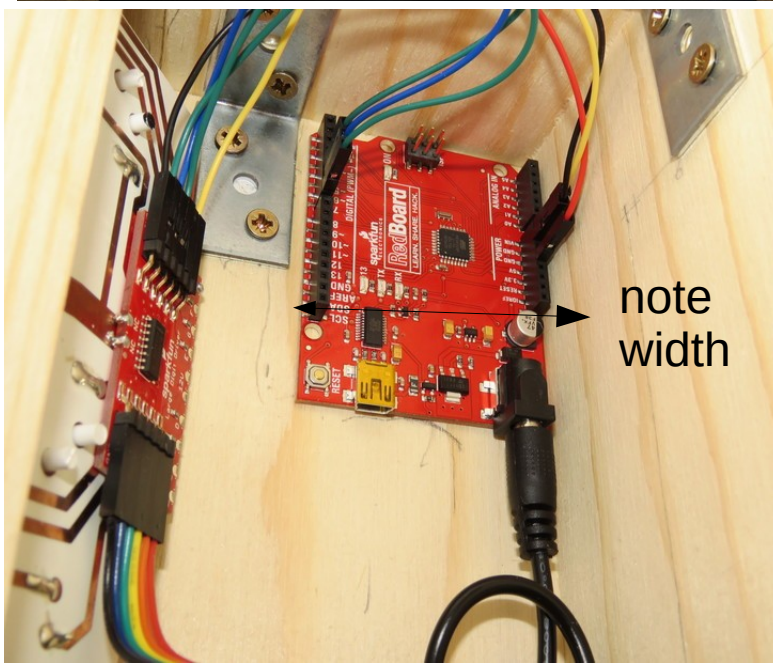
dont forget you need <https://www.sparkfun.com/products/9442> your contry plug, and keep the recomendend minimal 12VDC 600mA (I have 1000mA and shine great)



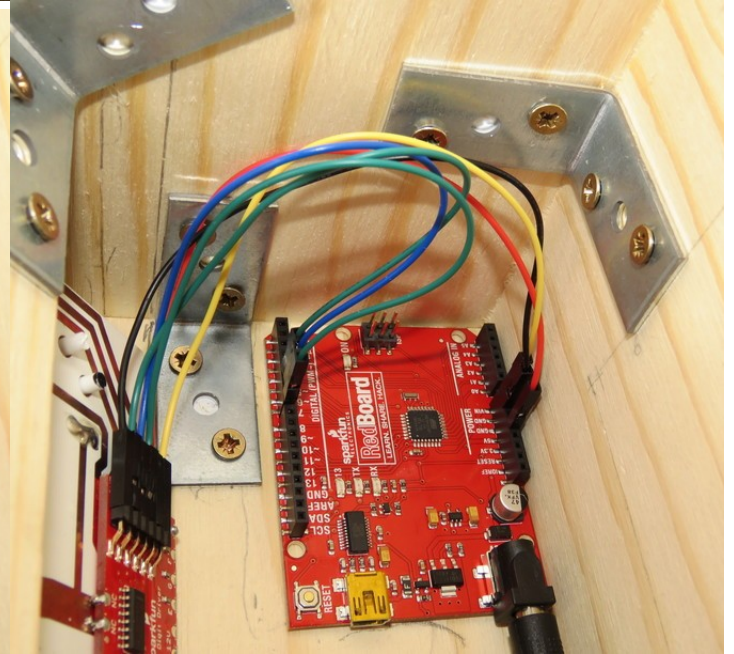
the depth of the digit (without plate) is aprox 1.3cm
 note it for page where "my glass break because of hole"



the digits are hanging
the "1x digits holder 48x15"
is aligned with top



note
width



digits are approx 2cm from top



pillars 1x1 cm
2x 21cm 1x46cm



triangular bottom **back** for glass
(before it was glued in, there was
hole

hole for
power

hole
for usb
cable

don't do this. lit is
useless.
The usb (not on
picture) and
power are
keeping it good
enough. Plug them
out, and pick up
arduino



pillars, so the
glass do not
press against
digits

hole for
power



top (50x20)



scale



other bolts (especially on back) are just through

this are only two places where bolt is tightened by nut



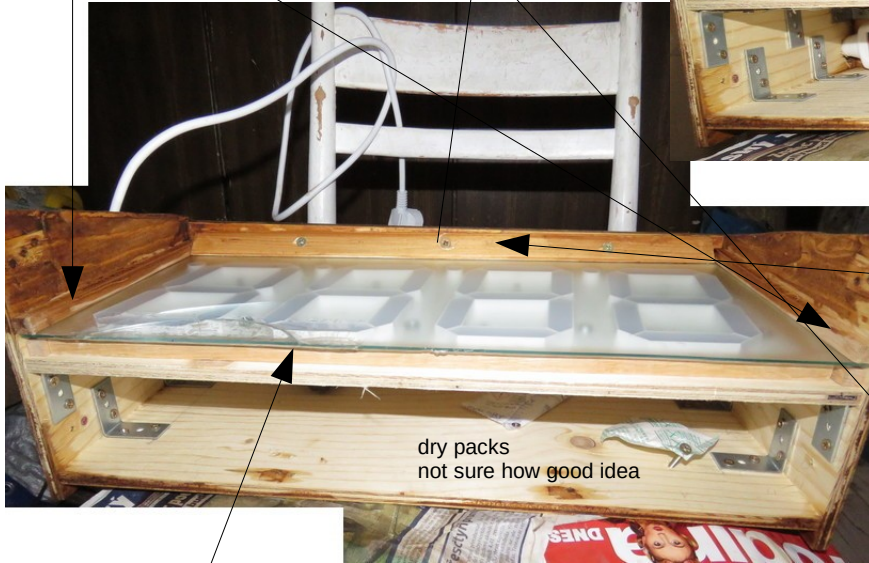
this are only two places where bolt is tightened by nut

other bolts (especially on back) are just through

glass and external penetration

second row of side pillars

do not press glass to strong!



dry packs
not sure how good idea

this hole was error, and my glass had break because of it. Glue something in



bottom
triangular pillar
top



box was completely oiled
by **PNZ exterior oil**



after completion, it is recommended to glue old **pvc** to top and let it be **overhanging**, so the bolts are covered



/*
Controlling large 7-segment displays
By: Nathan Seidle
SparkFun Electronics
Date: February 25th, 2015
License: This code is public domain but you buy me a beer if you use this and we meet someday (Beerware license).

The large 7 segment displays can be controlled easily with a TPIC6C594 IC. This code demonstrates how to control four display.

Here's how to hook up the Arduino pins to the Large Digit Driver

Arduino pin 6 -> CLK (Green on the 6-pin cable)
5 -> LAT (Blue)
7 -> SER on the IN side (Yellow)
5V -> 5V (Orange)
Power Arduino with 12V and connect to Vin -> 12V (Red)
GND -> GND (Black)

There are two connectors on the Large Digit Driver. 'IN' is the input side that should be connected to your microcontroller (the Arduino). 'OUT' is the output side that should be connected to the 'IN' of additional digits.

Each display will use about 150mA with all segments and decimal point on.
*/

```
//GPIO declarations
//=====
byte segmentClock = 6;
byte segmentLatch = 5;
byte segmentData = 7;
//=====

void setup()
{
  Serial.begin(9600);
  Serial.println("Large Digit Driver Example");

  pinMode(segmentClock, OUTPUT);
  pinMode(segmentData, OUTPUT);
  pinMode(segmentLatch, OUTPUT);

  digitalWrite(segmentClock, LOW);
  digitalWrite(segmentData, LOW);
  digitalWrite(segmentLatch, LOW);

  int x = 0;
  //x= 5200; //test
  while(1)
  {
    int second=x%60;
    int minute=x/60;
    int sd1=second%10;
    int sd2=second%10;
    int md1=minute%10;
    int md2=minute%10;
    //remember it is shifting, so first sent, is last in appearing row
    showNumber(sd2, false);
    showNumber(sd1, false);
    showNumber(md2, true);
    showNumber(md1, false);
    delay(999); //a bit faster is better then a bit slower

    x++;
    //x%= 120; //test, Reset x after 2minutes
    x%= 5400; //Reset x after 90minutes

    Serial.println(x); //For debugging
  }
}

void loop()
{
  //Test pattern
  //showNumber(8, true);
  //showNumber(8, false);
  //showNumber(8, true);
  //showNumber(8, false);

  //Takes a number and displays a number. Displays absolute value (no negatives)
  void showNumber(float value, bool dot)
  {
    int number = abs(value); //Remove negative signs and any decimals
    postNumber(number, dot);

    //Latch the current segment data
    digitalWrite(segmentLatch, LOW);
    digitalWrite(segmentLatch, HIGH); //Register moves storage register on the rising edge of RCK
  }

  //Given a number, or '-', shifts it out to the display
  void postNumber(byte number, boolean decimal)
  {
    // - A
    // - F B
    // - G
    // - E C
    // - D P

    #define a 1<<0
    #define b 1<<8
    #define c 1<<5
    #define d 1<<4
    #define e 1<<3
    #define f 1<<1
    #define g 1<<2
    #define dp 1<<7

    byte segments;

    switch (number)
    {
      case 1: segments = b | c; break;
      case 2: segments = a | b | d | e | g; break;
      case 3: segments = a | b | c | d | g; break;
      case 4: segments = f | g | b | c; break;
      case 5: segments = a | f | g | c | d; break;
      case 6: segments = a | f | g | e | c | d; break;
      case 7: segments = a | b | c; break;
      case 8: segments = a | b | c | d | e | f | g; break;
      case 9: segments = a | b | c | d | f | g; break;
      case 0: segments = a | b | c | d | e | f; break;
      case '-': segments = 0; break;
      case 'c': segments = g | e | d; break;
      case '-': segments = g; break;
    }

    if (decimal) segments |= dp;

    //Clock these bits out to the drivers
    for (byte x = 0; x < 8; x++)
    {
      digitalWrite(segmentClock, LOW);
      digitalWrite(segmentData, segments & 1 << (7 - x));
      digitalWrite(segmentClock, HIGH); //Data transfers to the register on the rising edge of SRCK
    }
  }
}
```

The code for arduino to create simple 90minutes, per second, counter.

Note that the large digits driver is very dummy, shift register.

So program is actually writing state (0/1) of last led of last digits first. The state of last-1 led of last digit and so on until state of second led of first digits, and first led of first digit last are written.

It is 8segment (7+dot) so one byte is for one number, so you really write 4bytes to the output.

The human readable postNumber method composing eight bits in byte so they represent 8states of eight leds's on/of. It is writing this single byte. So it is writing one digit (from its last led to first)