

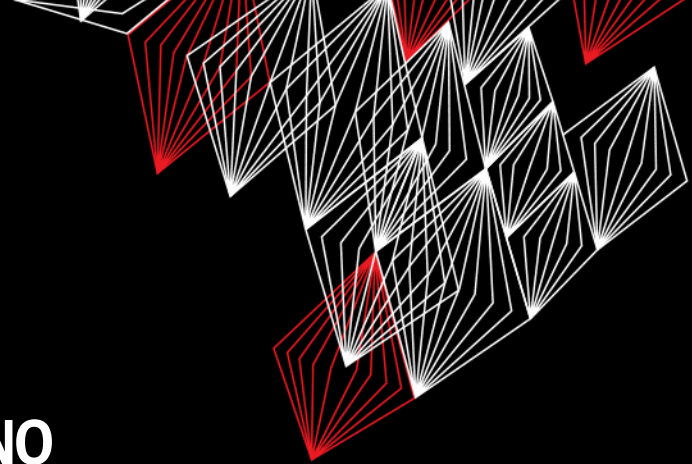
UNIVERSITY OF TWENTE.

# LEGO MINDSTORMS & ARDUINO

PRACTICAL SESSION 3



Part of **SmartProducts**





# LEGO MINDSTORMS & ARDUINO

## PRACTICAL SESSION 2

Fjodor van Slooten  
W241 (*Horst-wing West*)  
f.vanslooten@utwente.nl

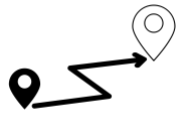


- Using Arduino modules: displays, sensors and Wifi
- Introduction to Blynk: control Arduino with an App
- Arduino programming - part 3
- Assignment

If you missed one of the mandatory sessions, a re-sit will be held Fri May 18th, 13:45-17:30h

# LAST WEEK

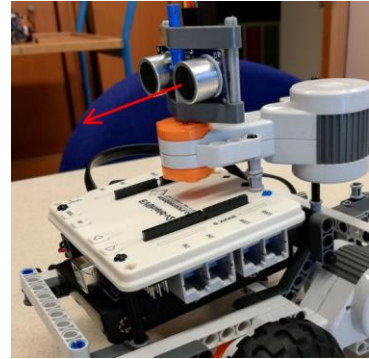
---



- 😊 Most of you were able to do at least 3 challenges
- 😞 These challenges were the simple ones...
  - Do you feel you are making progress?

Tip:

- If you download an example, open it in Arduino IDE, then first do *File > Save As*, to save it in your *Documents\Arduino* folder



# LCD DISPLAY WITH I2C BACKPACK

2 lines of text, bright backlight,  
blue screen, white text



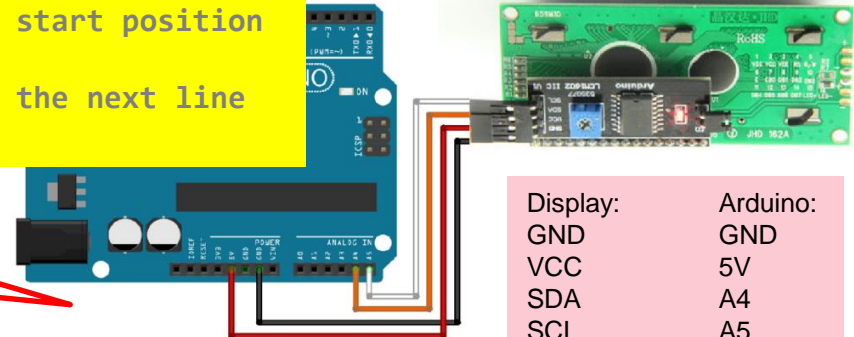
Type: HD44780

- Library: [Newliquidcrystal 1.3.5](#)
- Example: [LCD HD44780 i2c hello world example.ino](#)
- Change address: 0x27 or 0x3F

Examples that  
come with library  
do not work out-  
of-the-box

```
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7);  
// first parameter is address, can be 0x27, 0x3F or 0x38  
  
lcd.home(); // go to start position  
lcd.print("Hello, world!");  
lcd.setCursor(0,1); // go to the next line  
lcd.print("AppDev example");
```

Shares i2c connector (A4,A5) with  
EVShield, but works fine!



Display:	Arduino:
GND	GND
VCC	5V
SDA	A4
SCL	A5

# EXTRA LEGO LED DISPLAY

## CONNECT TO EVSHIELD

---

- 2 lines of text
- Connects to sensor port of EVShield (e.g. BAS1)

 Example:

[evshield\\_lego\\_display.ino](#) (requires added class **EVs\_Display**, in version as distributed in zip-file AppDev)

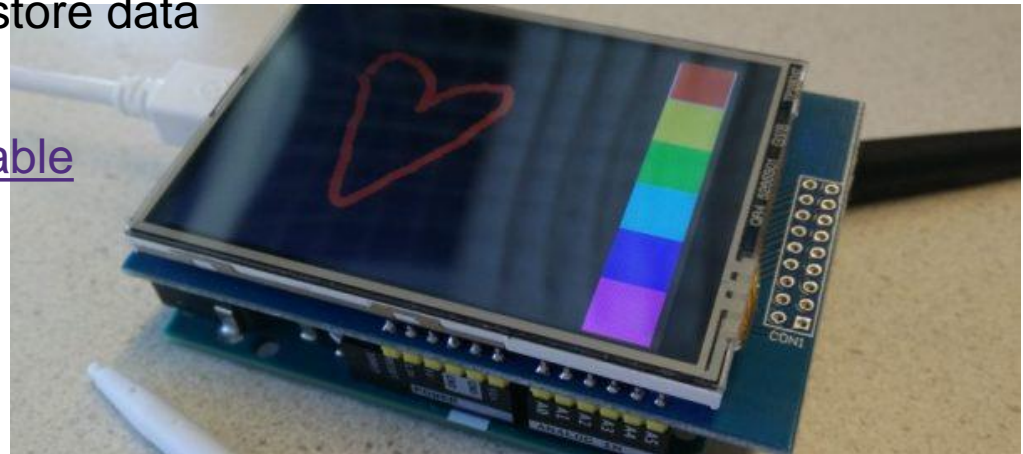


# MORE DISPLAYS

CAN BE BORROWED

---

- Color TFT Touch screen
- Can be combined with EVShield
- Has SD card socket, to store data (e.g. images)
- [Example sketches available](#)

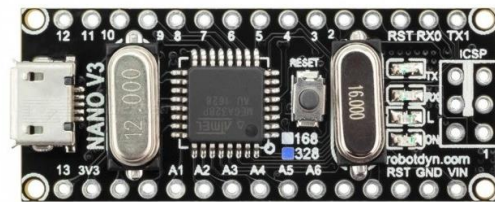




# ARDUINO NANO



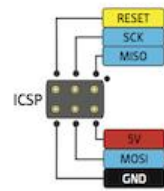
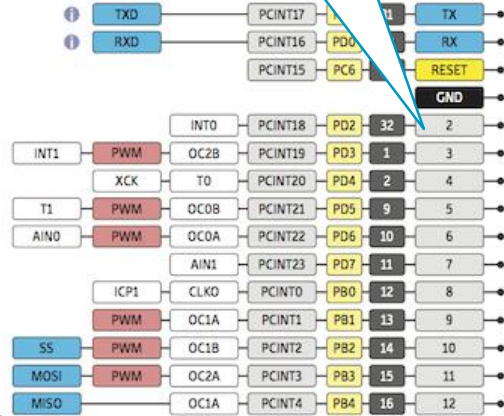
- Black or blue one



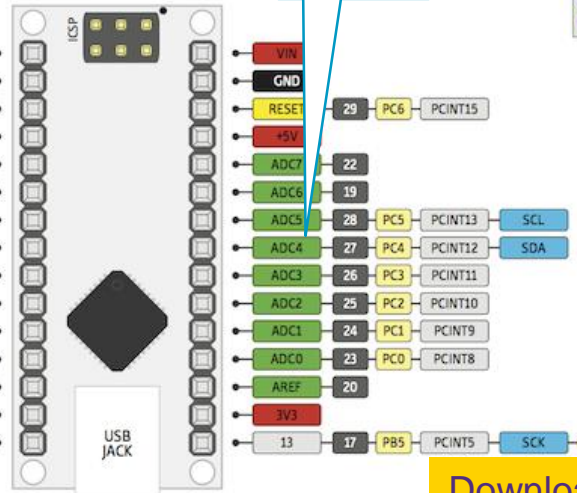
⚠ Absolute max per pin 40mA recommended 20mA  
 ⚡ Absolute max 200mA

D2, used by temperature sensor

Connected to one Arduino and used for USB program and communicating with it



A4 A5, used by display



LEGEND	
	GND
	POWER
	CONTROL
	PHYSICAL PIN
	PORT PIN
	ATMEGA328 PIN FUNC
	DIGITAL PIN
	ANALOG-RELATED PIN
	PWM PIN
	SERIAL PIN
	General Information
	Pay Attention
	No Really PAY ATTENTION
	LED

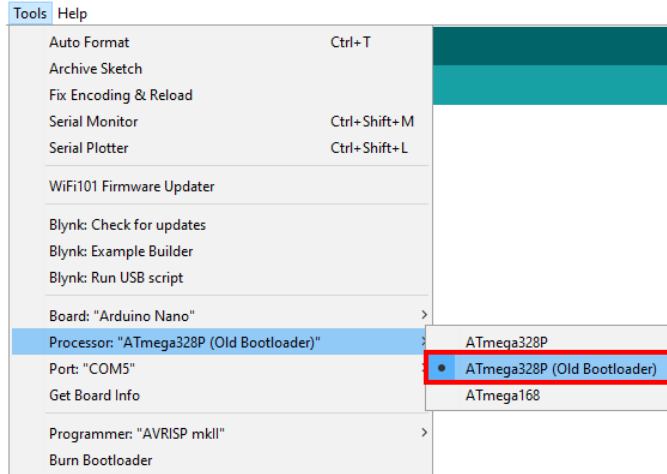
ⓘ On version 2 Analog Pins are reversed e.g. A0↔A7, A7↔A0

# ARDUINO NANO

## UPLOAD SKETCH



### ■ Error uploading?



An error occurred while uploading the sketch

```
avrdude: stk500_getsync() attempt 5 of 10: not in sync: resp=0x00
avrdude: stk500_getsync() attempt 6 of 10: not in sync: resp=0x00
avrdude: stk500_getsync() attempt 7 of 10: not in sync: resp=0x00
avrdude: stk500_getsync() attempt 8 of 10: not in sync: resp=0x00
avrdude: stk500_getsync() attempt 9 of 10: not in sync: resp=0x00
avrdude: stk500_getsync() attempt 10 of 10: not in sync: resp=0x00
An error occurred while uploading the sketch
```

**Select Old Bootloader!**



# TEMPERATURE & HUMIDITY

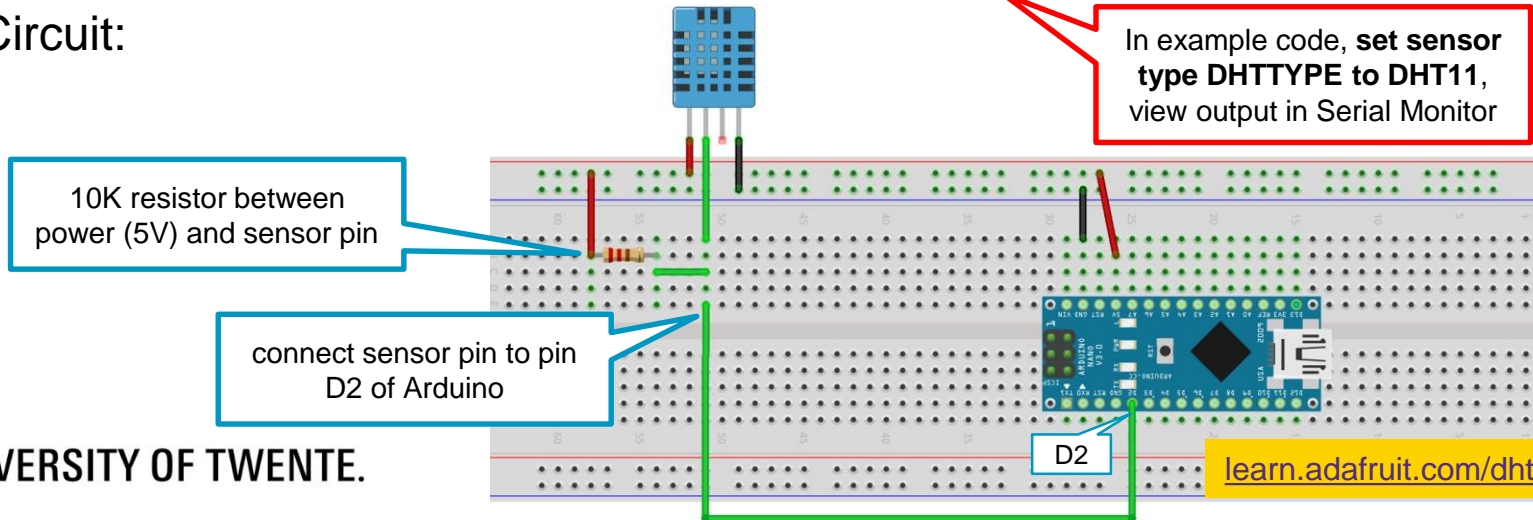


- DHT11 sensor
- If not installed yet, install 2 libraries: "Adafruit Unified Sensor" and "DHT library" (via *Sketch > Include Library*, search for the name)

Example: *File > Examples > DHT sensor library*, "DHT\_Unified\_Sensor"



- Circuit:



# TEMPERATURE & HUMIDITY

## SHOW OUTPUT ON DISPLAY

- Start with example “DHT\_Unified\_Sensor”
- Add display code

at top:

```
#include <LCD.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,2,1,0,4,5,6,7); // 0x27 or 0x3F
```

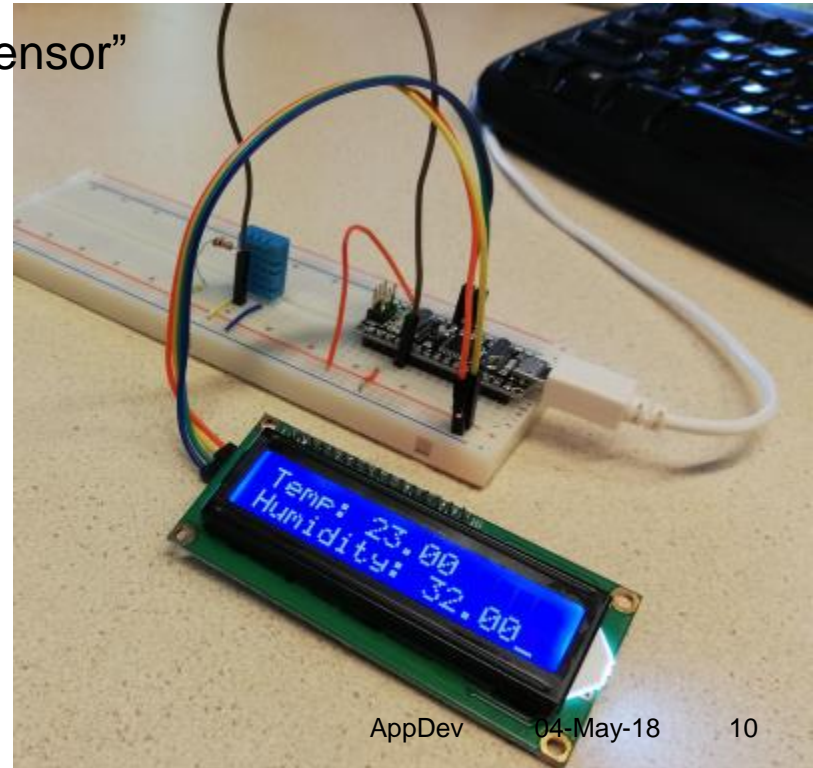
in setup():

```
// activate LCD module
lcd.begin(16,2); // for 16 x 2 LCD module
lcd.setBacklightPin(3,POSITIVE);
lcd.setBacklight(HIGH); // turn on backlight
```

in loop(): find spot where temperature is printed, add:

```
lcd.home(); // set cursor to 0,0
lcd.print("Temp: ");
lcd.print(event.temperature);
```

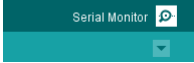
... repeat for humidity



# COMMUNICATION: SERIAL CONNECTION

Speed, can be 115200 or other

```
Serial.begin(9600);  
Serial.print("Temperature: "); Serial.println(temp);
```

- USB cable
  - Serial Monitor in Arduino IDE 
  - Another App, e.g. your own Java App: Example in Appendix of this mornings [Java assignment 3](#)
- Wired (via pins) to another device (e.g. another Arduino)
  - [Using RX/TX pins](#) (also used by USB!)
  - [Using any other pins](#)
- Wireless e.g. via Wifi or Bluetooth module

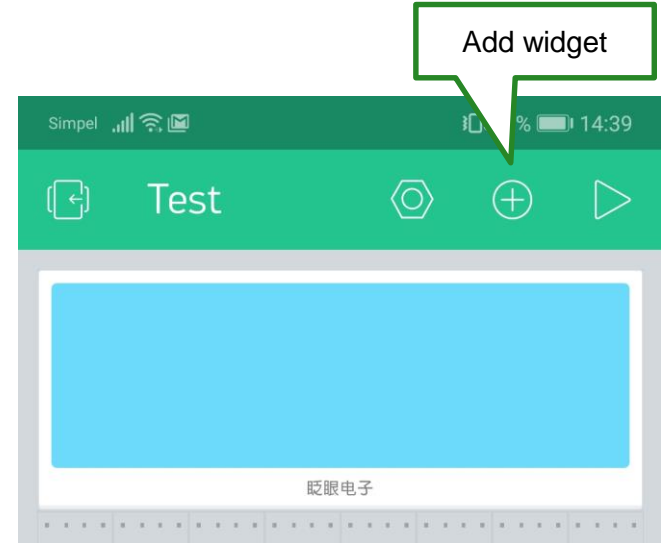
# BLYNK APP: ADD DISPLAY



## Step 1/5

- [Getting-started tutorial](#) (use Blynk via USB cable)
- Get Blynk App (search Play Store)
- Start project (device: Arduino Nano, connection type USB)
- Add a display

Learn more: [docs.blynk.cc/#widgets-displays-lcd](https://docs.blynk.cc/#widgets-displays-lcd)



# BLYNK: USB CONNECTION

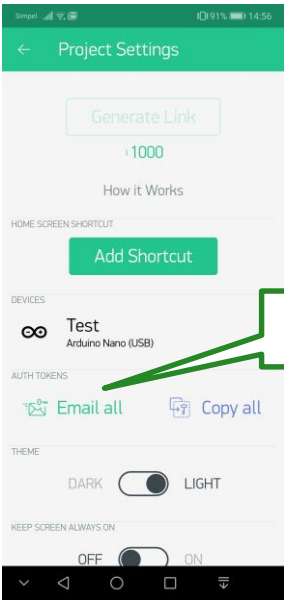
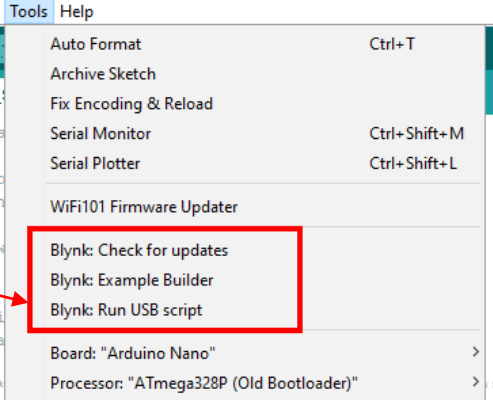
(NO WIFI)



## Step 2/5

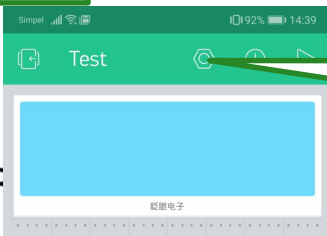
- Start with *File > Examples > Blynk > Boards\_USB\_Serial > Arduino\_Serial\_USB*
- Check Blynk-entries in tools menu: if not there, copy Arduino\tools folder from AppDev zip-file to Documents\Arduino and restart Arduino IDE
- Fill in Auth Token:

Save As ...



Get Auth Token

```
34 #include <BlynkSimpleStream.h>
35
36 // You should get Auth Token in the Blynk App.
37 // Go to the Project Settings (nut icon).
38 char auth[] = "cbbc4b13f8d4135a930a91f0881d7be";
```



Go to project settings

# BLYNK: PRINT TO DISPLAY

## CONNECTION VIA USB



### Step 3/5

Learn more: [docs.blynk.cc/#widgets-displays-lcd](https://docs.blynk.cc/#widgets-displays-lcd)

- Add code to sketch for Blynk LCD widget:  
at top:

```
WidgetLCD lcd(V1); // this is a display widget in the Blynk Userinterface
```

at end of setup():

```
lcd.clear(); // clear the display in the Blynk App  
lcd.print(0,0, "Hello Blynk");
```

- Close Blynk tool
- Upload sketch
- Run *Tools > Blynk: Run USB script:*

```
Blynk  
Ensure that Serial Monitor and Plotter are closed when using this tool.  
It uses same port and speed as Serial Monitor  
  
Connecting device at COM5 to blynk-cloud.com:80...  
OpenC0C("\\.COM5", baud=9600, data=8, parity=no, stop=1) - OK  
Connect("blynk-cloud.com", "80") - OK  
InOut() START  
DSR is OFF
```

```
Blynk  
Ensure that Serial Monitor and Plotter are closed when using this tool.  
It uses same port and speed as Serial Monitor  
  
Connecting device at COM5 to blynk-cloud.com:80...  
OpenC0C("\\.COM5", baud=9600, data=8, parity=no, stop=1) - OK  
Connect("blynk-cloud.com", "80") - OK  
InOut() START  
DSR is OFF
```

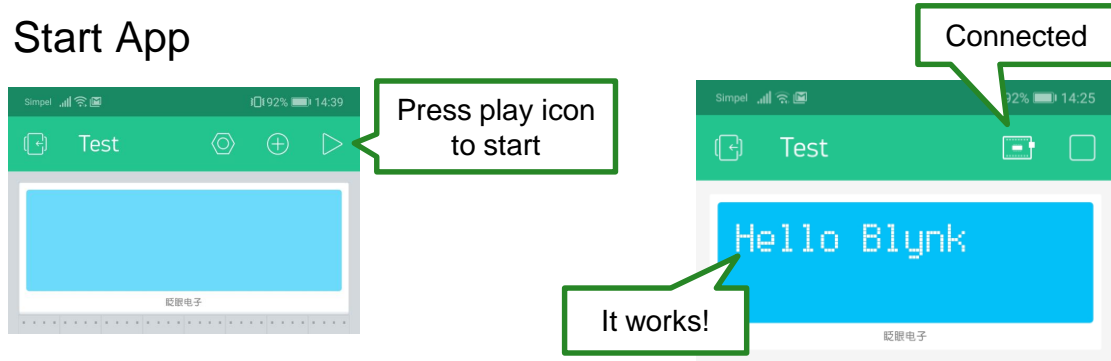
# BLYNK: USB CONNECTION

(NO WIFI)



## Step 4/5

- Start App





# BLYNK: ADD SENSOR CODE

## CONNECTION VIA USB



### Step 5/5

Learn more: [docs.blynk.cc/#widgets-displays-lcd](https://docs.blynk.cc/#widgets-displays-lcd)

- Use code from example "DHT\_Unified\_Sensor"

at top:

```
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>

#define DHTPIN 2 // Pin which is connected to the DHT sensor
#define DHTTYPE DHT11

DHT_Unified dht(DHTPIN, DHTTYPE);
```

in setup():\*

```
// Initialize sensor
dht.begin();
```


\* Remove "Hello Blynk" print-statement from setup()

in loop(): after `Blynk.run()`; add:

```
// Get temperature event and print its value
sensors_event_t event;
dht.temperature().getEvent(&event);
if (isnan(event.temperature)) {
  Serial.println("Error reading temperature!");
}
else {
  lcd.print(0,0,"Temp: ");
  lcd.print(10,0,event.temperature);
}
delay(5000); // delay 5 seconds
```

... repeat for humidity

Run sketch:

- Close Blynk USB Script window
- Upload sketch 
- Run *Tools > Blynk: Run USB script*
- Check Blynk App



[Display temperature as a nice labeled value?](#)



# BLYNK: WARNING

## USE EVENTOR AND NOTIFICATION WIDGET



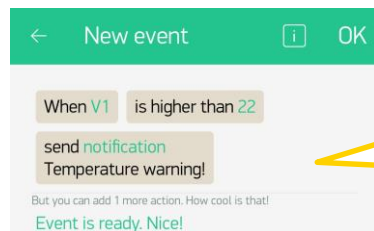
- Add Eventor and Notification widget,
- Configure Eventor:

Learn more:

[docs.blynk.cc/#widgets-other-eventor](https://docs.blynk.cc/#widgets-other-eventor)  
[docs.blynk.cc/#widgets-notifications-push-notifications](https://docs.blynk.cc/#widgets-notifications-push-notifications)



Eventor

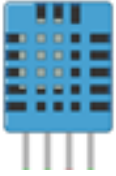


Drawback: generates lots of events... might be better to do via code, and add a limit

- Add code:

```
// send temperature to Virtual Pin 1 in the App,  
// to be able to process it in the Eventor:  
Blynk.virtualWrite(V1, event.temperature);
```

Alternative: send notification from code in Arduino sketch:  
`if (event.temperature>24) Blynk.notify("Temperature warning");`



# ASSIGNMENT

## FOR TODAY'S PRACTICAL SESSION

This assignment consists of 3 slides:

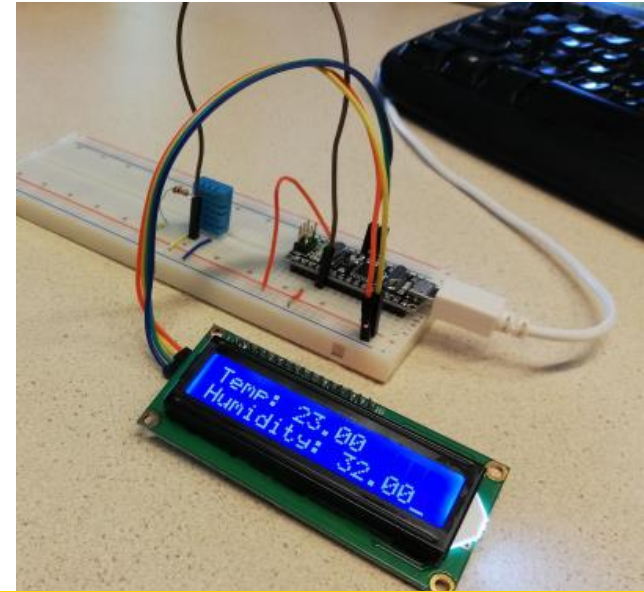
1/3

- Build a connected temperature & humidity sensor

If you prefer to finish challenges of last session, that is also OK.

If you missed one of the mandatory sessions, a re-sit will be held Fri May 18th, 13:45-17:30h

If your kit is missing pieces/materials, you can get replacements from teacher!



slides @ [vanslooten.com/appdev](https://vanslooten.com/appdev)

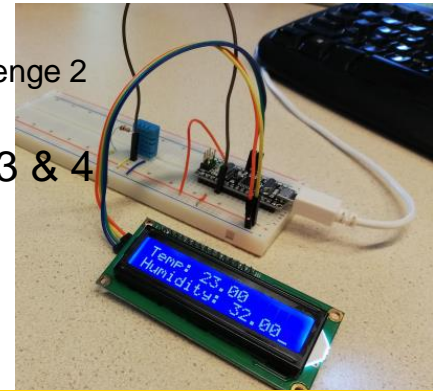
# ASSIGNMENT

## CONSTRAINTS

This assignment consists of 3 slides:

2/3

- Do this in **groups of 2 or 3** (sit together with project group at one table)
- Group box contains: 3 EL-kits with Arduino Nano, 1 Arduino Uno-kit
- You can do this assignment with any Arduino (Nano or Uno)
- If you use the Uno, you can add EVShield + Lego Display
- 1 temperature sensor in Arduino kit: **get more from teacher**
- You have only 2 displays (per project group): 1 couple skips challenge 2 and does other challenge (3, 4, 5) ... or borrows touch screen
- Arduino needs to be connected (USB or Wifi) for challenges 3 & 4



# ASSIGNMENT: CHALLENGES

This assignment consists of 3 slides:

3/3

Not enough  
displays? Skip this

1. Build a temperature sensor & view output in Serial Monitor
2. Add a LCD display: display both temperature & humidity
3. Connect the Arduino to laptop (USB cable): display temperature & humidity in the Java App you made this morning
4. Use Blynk to display the temperature on your phone (let Blynk communicate via either USB or Wifi)
5. Create a warning if the temperature reaches a limit (e.g. 24°).  
Can be:
  - A LED on the breadboard (add a LED)
  - Warning in the Java app
  - Warning in the Blynk app