

# Serial io

---

Using serial communication to talk to the Arduino  
Enable native scrolling

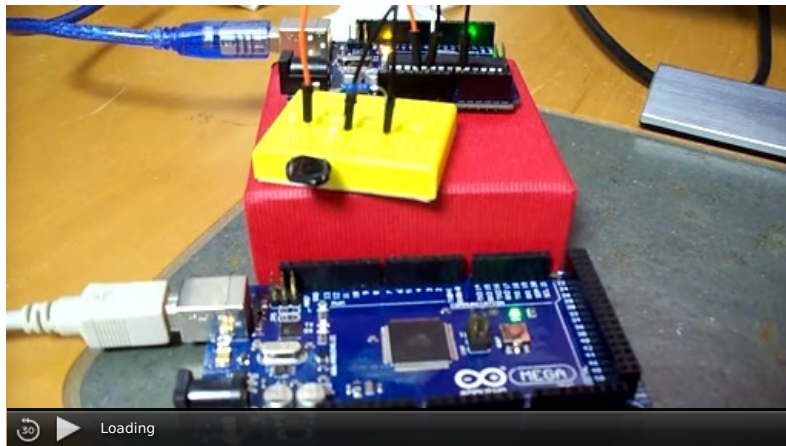
1/22

*But first:*  
solutions to the last assignment

2/22

## Morse tree

---



...

The code above sends some special characters to test the tree based decoder

3/22

## Morse tree

---

```
1 struct morse_tree tree_N_with_donald_hair =  
2   {"N", NULL, NULL};  
3 struct morse_tree tree_funny_hat_G =  
4   {"G", NULL, NULL};  
5 struct morse_tree tree_tick =  
6   {"'", NULL, NULL};
```

...

The code above receives the morse code and decodes it

4/22

# LED Tree

---

```
1 uint32_t now= millis();
2 if(now >= next_tree) {
3   ...
4   cur_node= random(2) ?
5     cur_node->child_l : cur_node->child_r;
6   digitalWrite(cur_node->pin, HIGH);
7   ...
8 }
```

...

Uses millis() instead of delay() to meet timing requirements

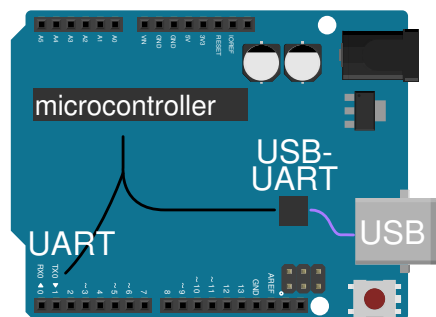
5/22

*Back to serial io*

6/22

## The serial port

---

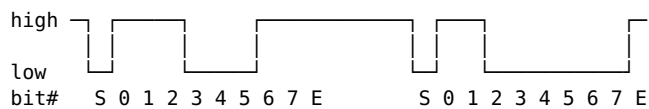


Pins 0(RX) and 1(TX) are connected to an USB-Serial converter

7/22

## The serial port

---



UART (Serial) is a simple communication protocol  
it uses one wire to receive data (RX) and one wire to transmit data (TX)

8/22

# Serial.read()

---

We used `Serial.print` and `Serial.println` for Arduino ► Computer communication (on the TX line)

Now we will use `Serial.read` for Computer ► Arduino communication (on the RX line)

`Serial.read()` returns a single character or -1 if no character was received

9/22

## Echo

---

```
1 void loop() {
2   char buffer[256]= {'\0'};
3   /* TODO */
4   Serial.print(buffer);
5 }
```

...

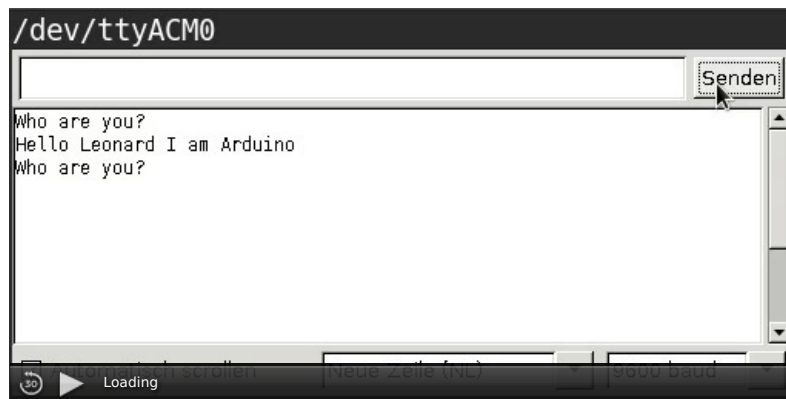
The code linked above contains a `read_line` function that reads a complete line from the serial port

10/22

## Echo

---

use `read_line` to enable conversations like the following:



*Hint:* Configure serial monitor to send newline characters

11/22

## resrever txet

---

```
1 void reverse_string(char *orig, char*rev)
2 {
3   size_t orig_len= strlen(orig);
4   size_t rev_idx= 0;
5   size_t fwd_idx= orig_len-1;
6
7   // TODO
8 }
```

...

The code above contains a broken `reverse_string` function

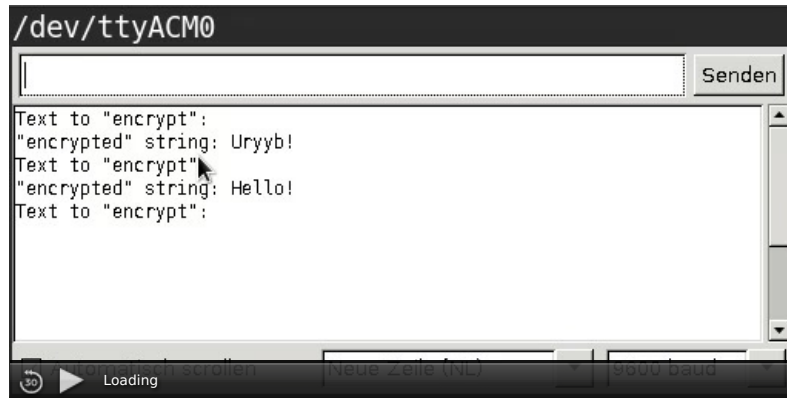
12/22



# rot13 pelcgb

---

Fix the rot13\_string function to enable conversations like the following:



16/22

## Regular expressions

---

[Mm][Aa]{1,10}(tch|gic)

Maaaagic matching machine

17/22

## Regular expressions

---

Michael Corner, @edgesncorners  
Dean Thomas, @magic\_dean  
Sue Li, @sureli  
Demelza Robins, dem@magic.com  
Susan Bones, sbones@minimagic.gov.uk

...

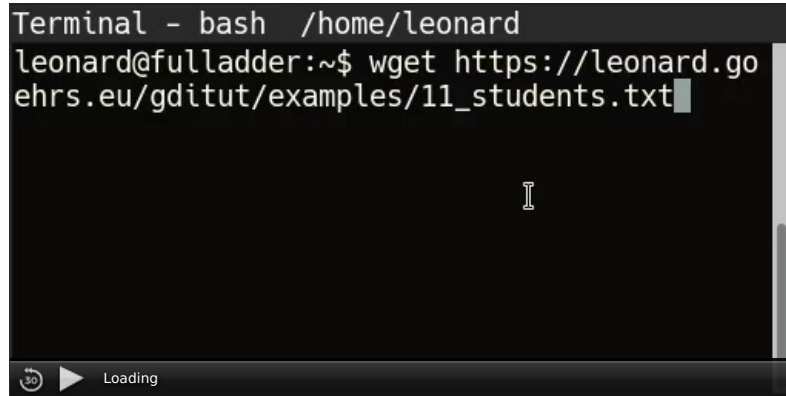
The list above contains Names and different kinds of contact informations

18/22

# Terminal basics

---

Open a terminal/console/powershell window

A terminal window with a black background and white text. The prompt is 'Terminal - bash /home/leonard'. The user has entered the command 'leonard@fulladder:~\$ wget https://leonard.goehrs.eu/gditut/examples/11\_students.txt'. A cursor is visible at the end of the command. At the bottom of the terminal, there is a 'Loading' indicator with a play button icon and a '30' timer.

```
Terminal - bash /home/leonard
leonard@fulladder:~$ wget https://leonard.goehrs.eu/gditut/examples/11_students.txt
```

Use wget to download the file linked in the previous slide

19/22

## grep

---

On Linux/macOS the [grep](#) command can be used to find lines in files that match a regular expression

Windows users can try the [sls](#) command or a web based regular expression tester

20/22

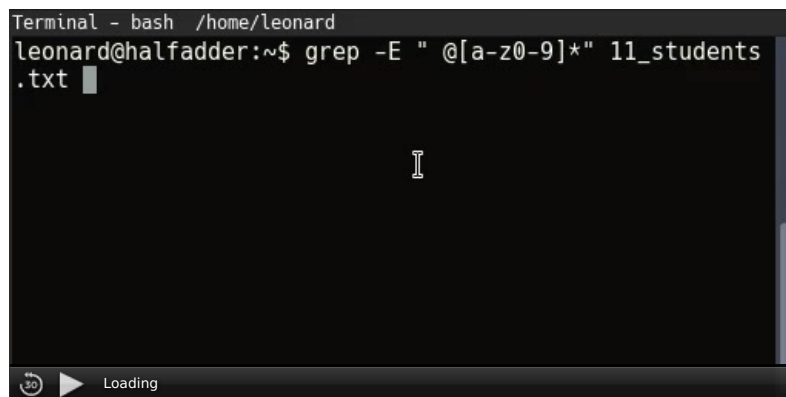
## grep

---

The general syntax is

Linux/macOS: `$ grep -E "regular expression" filename`

Windows: `PS C:\> sls "regular expression" filename`

A terminal window with a black background and white text. The prompt is 'Terminal - bash /home/leonard'. The user has entered the command 'leonard@halfadder:~\$ grep -E "[a-z0-9]\*" 11\_students.txt'. A cursor is visible at the end of the command. At the bottom of the terminal, there is a 'Loading' indicator with a play button icon and a '30' timer.

```
Terminal - bash /home/leonard
leonard@halfadder:~$ grep -E "[a-z0-9]*" 11_students.txt
```

The example above searches for all lines that contain a twitter handle as contact info

21/22

# grep

---

Using the lecture slides and other resources write regular expressions that match on lines that contain ...

- ... an email address
- ... a telephone number
- ... an UK based telephone number
- ... surnames between 1 and 5 characters long
- ... email address on a .uk domain