Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Arduino based GUI for Touchscreen Displays

Helen Schloh, B.Sc. Informatics

Supervisor: Andreas Mäder
Second Supervisor: Bernd Schütz
Submission Date: 31.3.2021

16.02.2021

Outline

- Research Question
- Definitions
- Hardware
  - Setup
  - Touchscreen
  - Arduino
  - Other Devices
- Software
  - Touch Detection
  - Display
  - Event handling
  - Simulation
- Evaluation
- Outlook

Research Question

What am I researching and why?

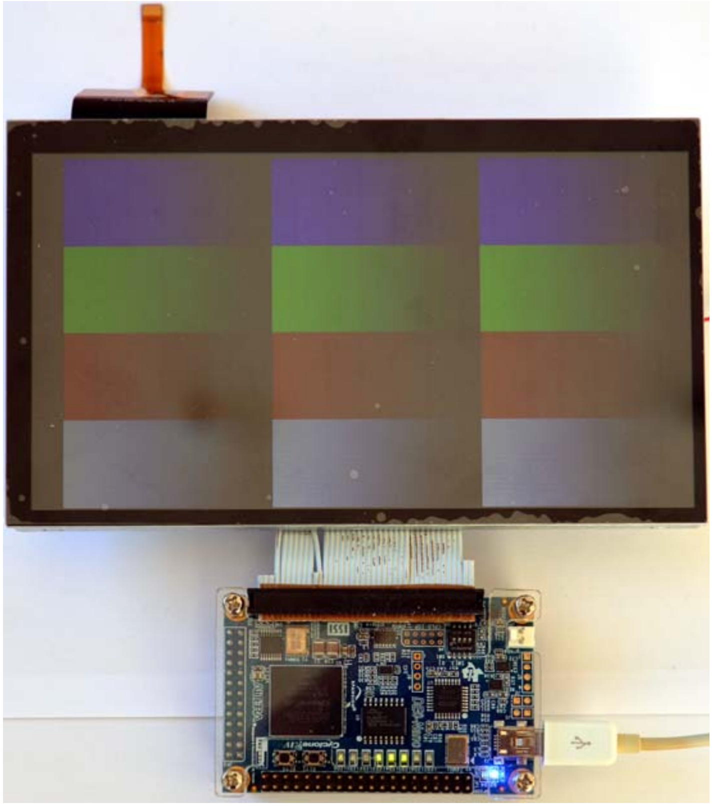
- Research setup
  - Develop GUI library for given Touchscreen Display and Arduino Due
- Purpose
  - Using the Display during ES exercises
  - Investigate performance of setup
  - Explore use cases
- Goal
  - Detect Gestures reliably
  - Draw GUI Elements on Display
  - Maximize Quality of speed and memory usage
  - Derive recommendations for using the GUI library

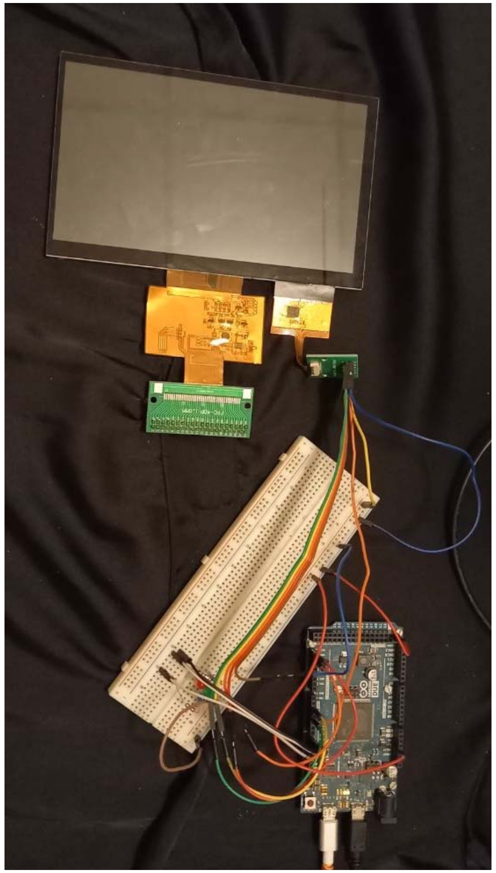
Definitions

- I²C
  - Communication protocol
  - Synchronous
  - Serial
- UART (Universal asynchronous receiver and transmitter)
  - Hardware Device for Communication
  - Asynchronous
  - Serial
- (Hardware) Interrupt:
  - Processor stops all tasks to execute interrupt routine

Hardware

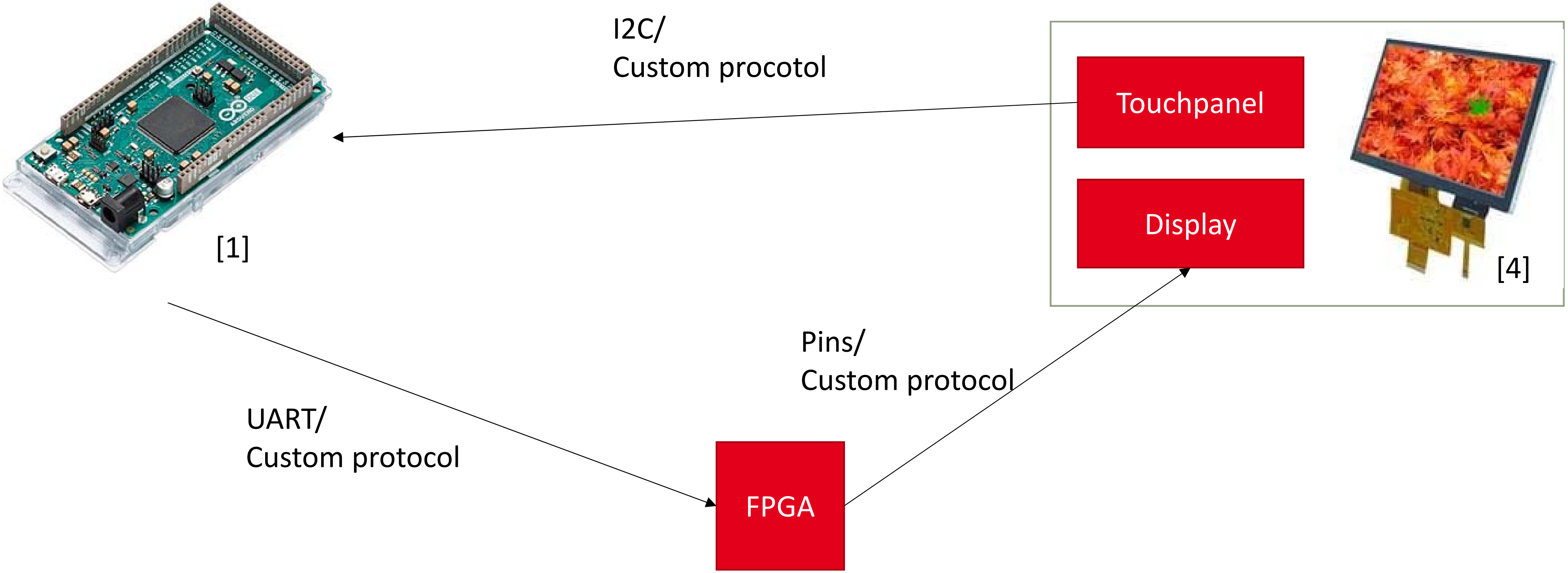
Setup





Hardware

What does the System look like?



I2C/
Custom procotol

Touchpanel

Display

[1]

[4]

Pins/
Custom protocol

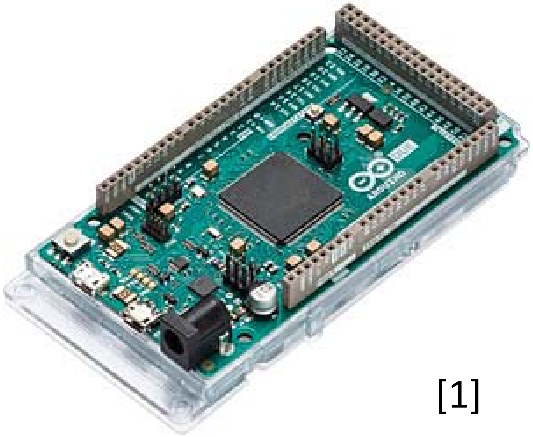
UART/
Custom protocol

FPGA

Hardware

Arduino Due

- Microcontroller Board
  - 84 MHz clock
- Serial Communication
  - I^{2}C : 100 KHz
  - UART
    - Serial: (max 115200 bps)
    - SerialUSB: (max 480 Mbps)
- Memory
  - 512 kB ROM
  - 64,32 kB RAM

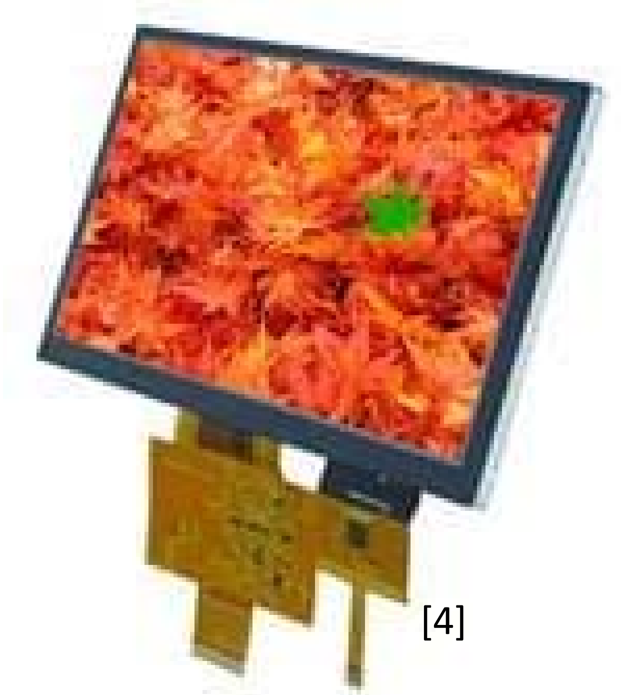


[1]

Hardware

Touchscreen

- EA TFT070-84ATS Touchscreen
- Display
  - 800x480 RGB
  - 33,3 MHz parallel Interface
  - Cannot be sent by Arduino
- Touch Panel
  - 1792x1024 dots
  - I²C Schnittstelle



[4]

Other Devices

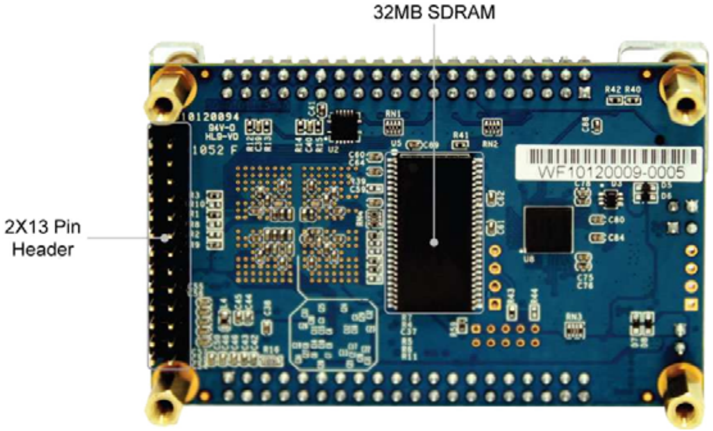
Windows PC and FPGA

- Windows PC:

  Acer Swift 1

[3]

- FPGA
  - DE0-Nano Development Board
  - 50 MHz clock



[2]

Software

Software

Architecture

- Object oriented C++
  - OO-metaphor fits GUI Element nesting
  - ES students know Java Swing
  - Arduino runs with C++

Detecting Gestures
Modeling finite Automaton as Class

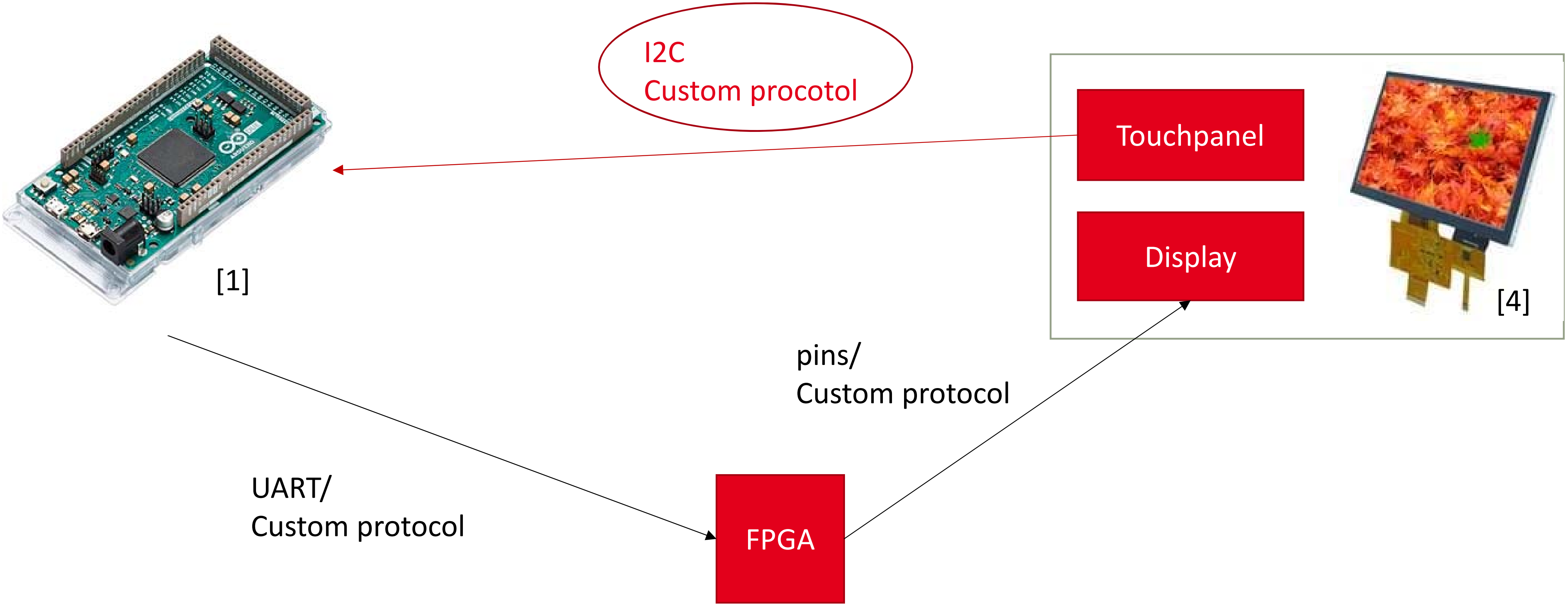
Display GUI
Develop GUI Elements as Classes

Handle Touch Events
Implement Listener Pattern

Software

Touch Detection

Interface - Touch



I2C
Custom procotol

Touchpanel

Display

[4]

[1]

pins/
Custom protocol

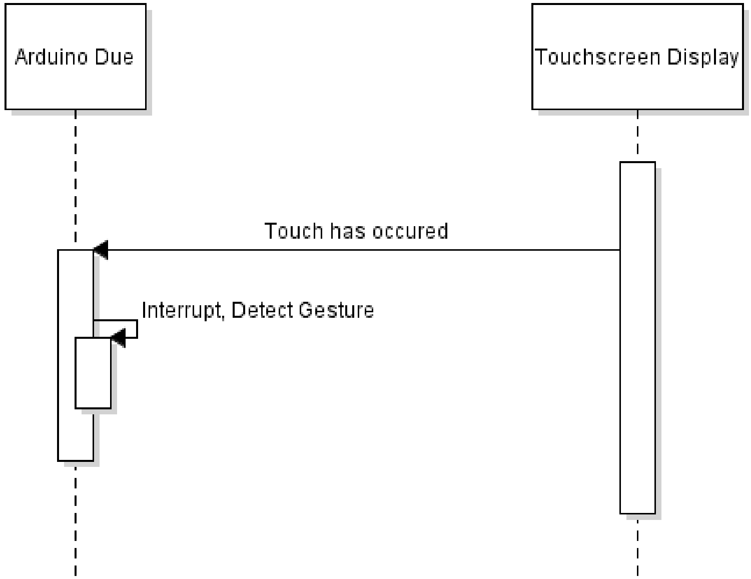
UART/
Custom protocol

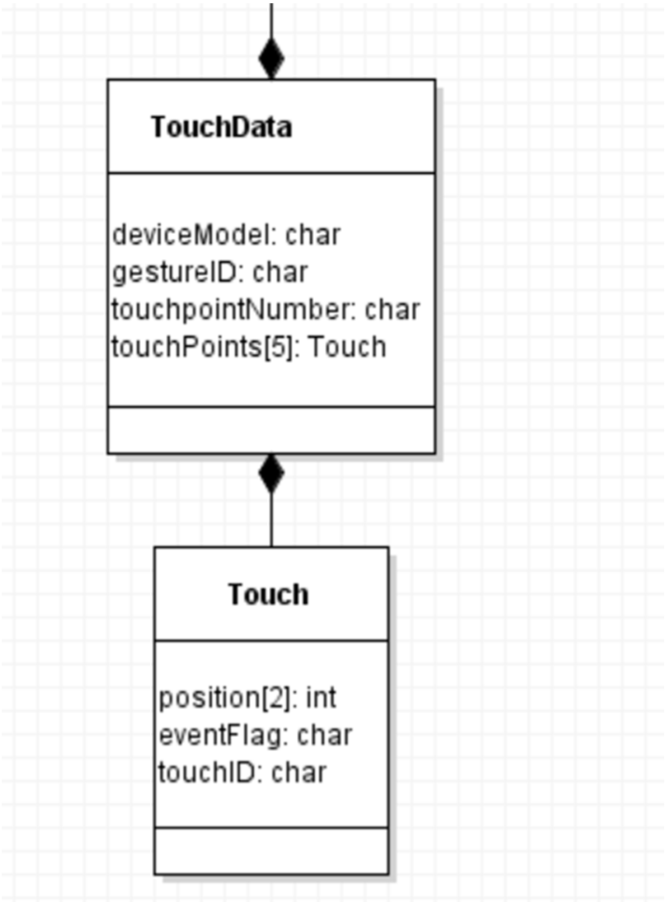
FPGA

Touch Recognition

How is Touch Information retrieved?

- Attach interrupt on touch occurring
- Retrieve touch information




TouchData

deviceModel: char
gestureID: char
touchpointNumber: char
touchPoints[5]: Touch

Touch

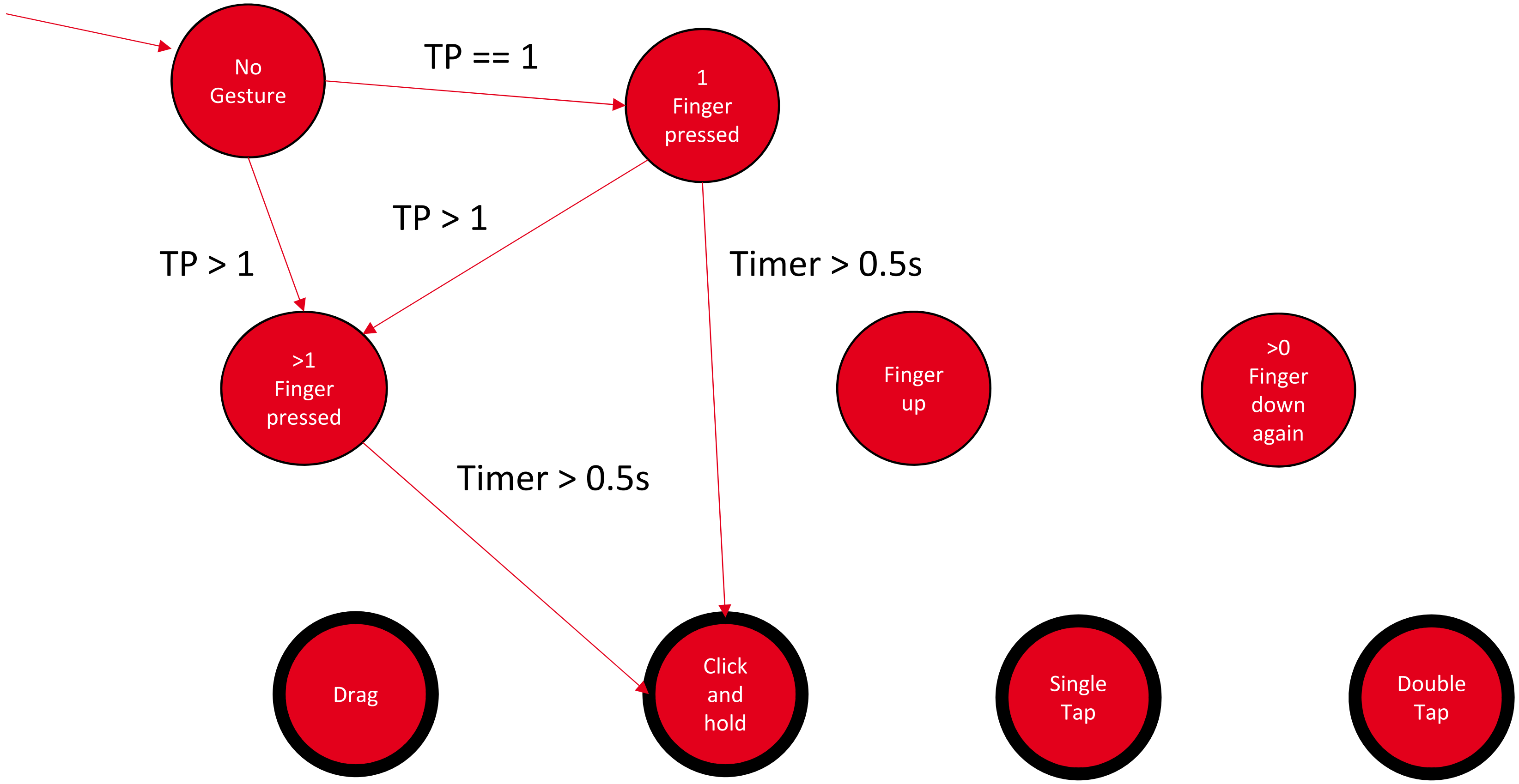
position[2]: int
eventFlag: char
touchID: char

Touch Recognition

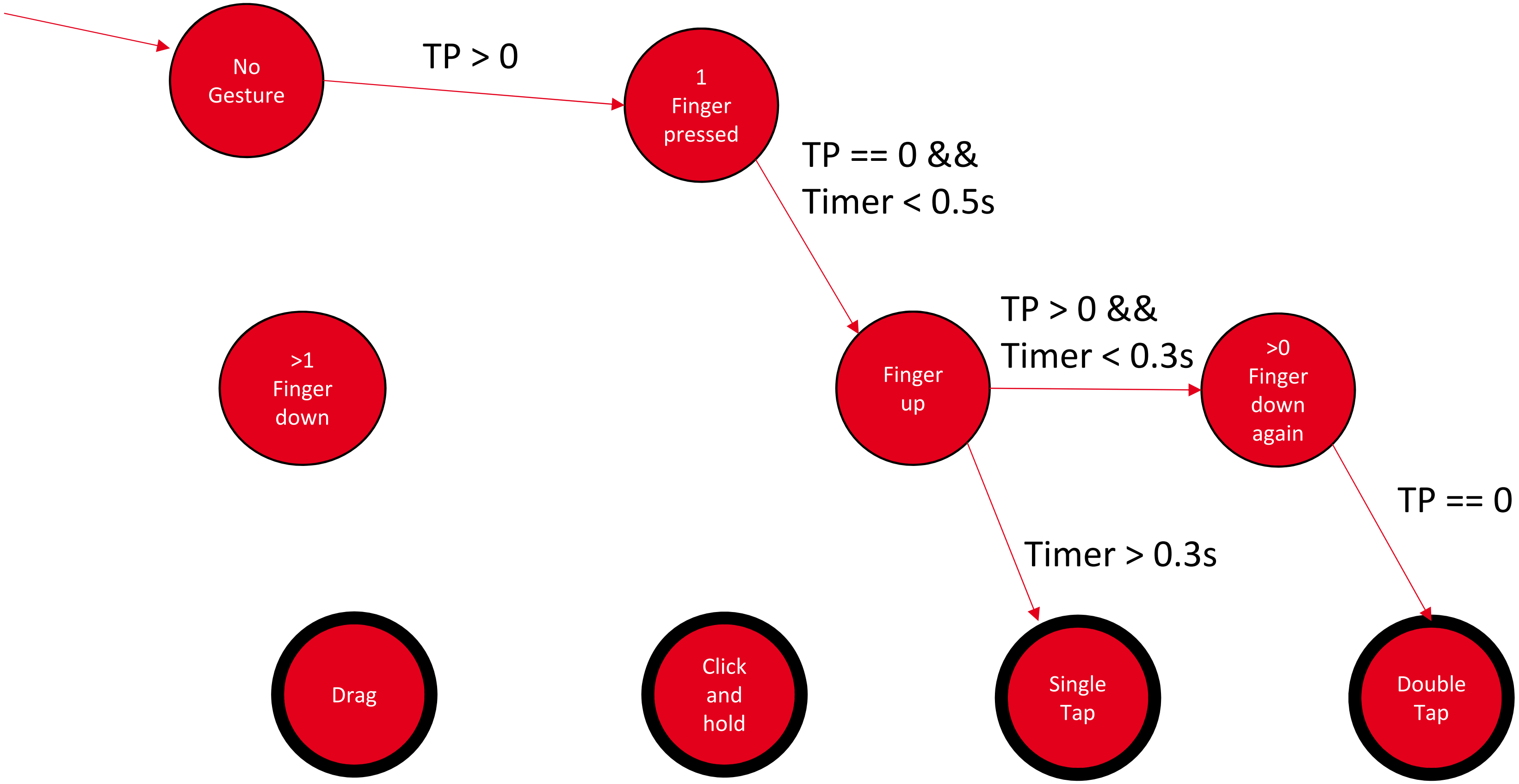
How are Touch Gestures recognized?

- Modeling Finite Automaton
  - Class as Automaton
  - Functions as States
  - Function calls for transition
- Change of state depending on
  - Number of Touchpoints (Fingers on Touchpanel): TP
  - Time passed since entering state: Timer
  - Position of Touches: x[n],y[n], n-th Touch

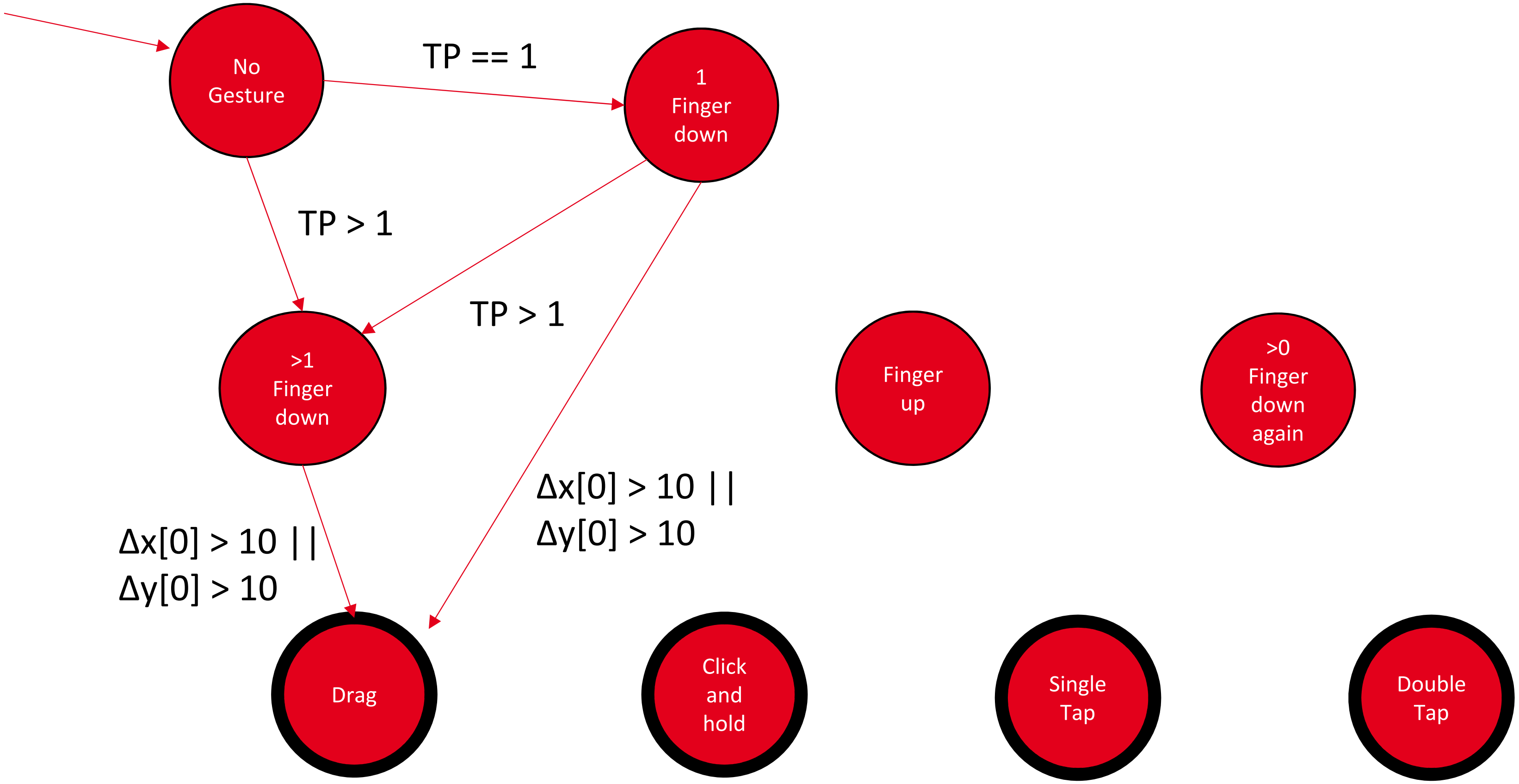
Touch Detection - Gestures



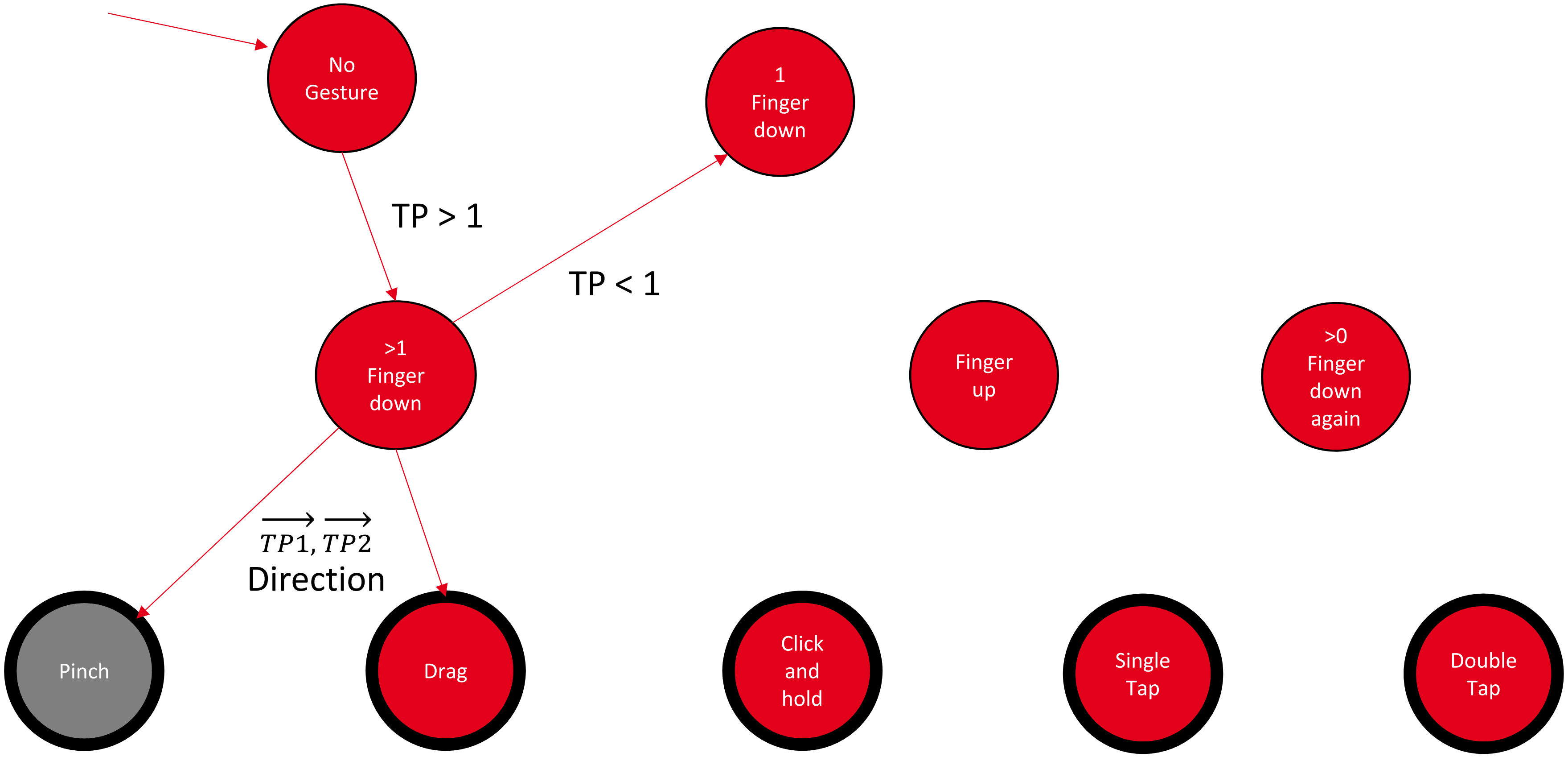
Touch Detection - Gestures



Touch Detection - Gestures



Touch Detection – Multitouch Gestures

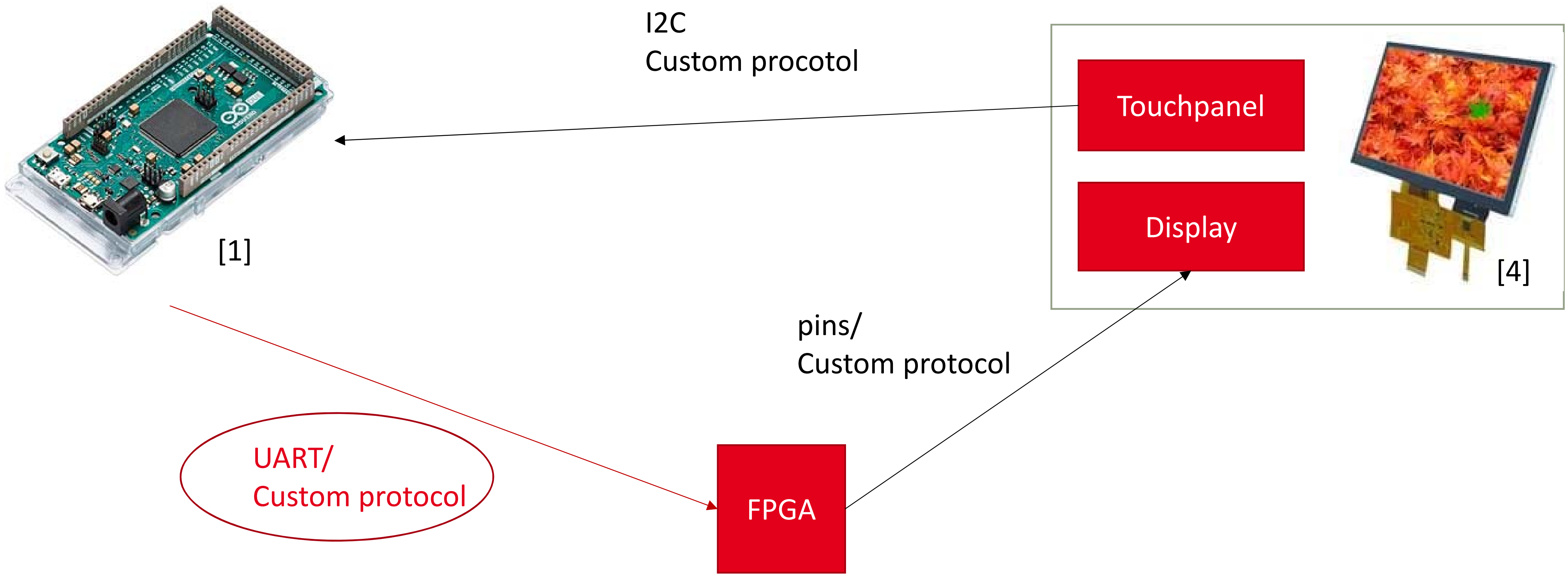


Software

Display

Interface – Display

How are GUI Elements displayed?



[1]

I2C
Custom procotol

Touchpanel

Display

[4]

pins/
Custom protocol

UART/
Custom protocol

FPGA

Interface – Display

What does the Arduino communicate to the FPGA?

Description	Opcode [8 bit]	Body
Clear Display	0x0	-
Set drawing Position	0x1	x [10 bit] y [10 bit]
Set drawing Color	0x4	Red [8 bit] Green [8 bit] Blue [8 bit]
Set pixel, Increment drawing position	0x2	-
Draw bitmap	0x3	Width [10 bit] Height [10 bit] Bitmap [(width*height) bit]

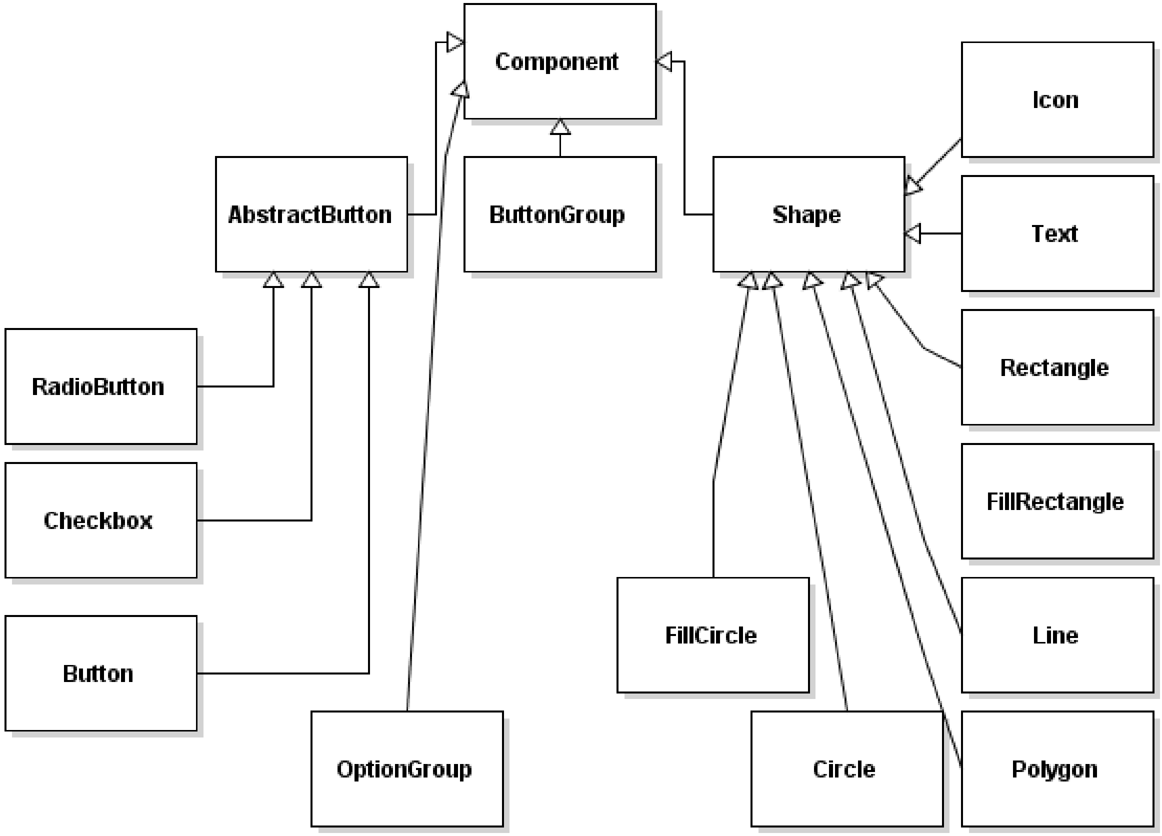
Interface – Display

Drawing Bitmaps

0	1	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	0	0	1	1	0	0	0
0	0	0	0	1	1	0	0
0	0	0	0	1	1	1	0
0	0	0	0	1	1	0	0
0	0	0	1	1	0	0	0
0	0	1	1	0	0	0	0
0	1	1	0	0	0	0	0

Overview of GUI Elements

Which Elements were implemented?

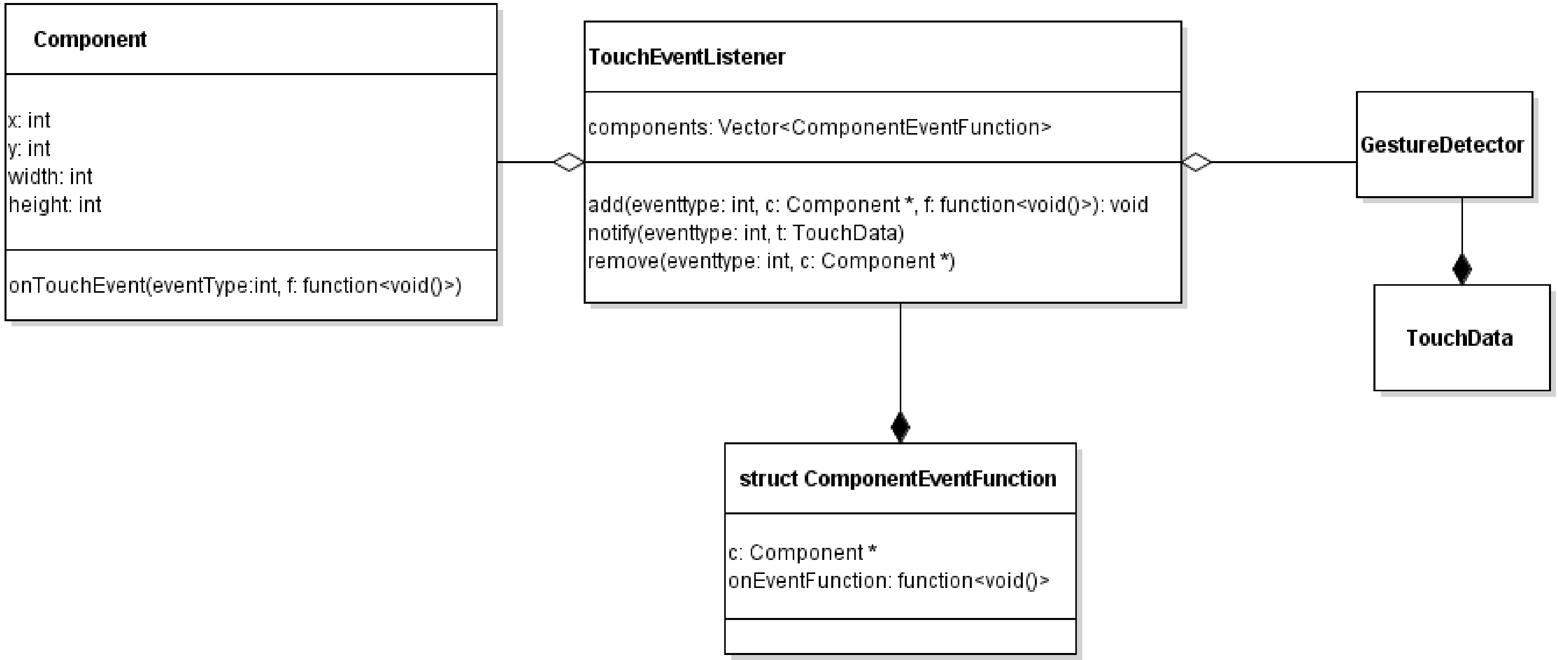


Software

Event Handling

Event handling

How does the Arduino react to Touch Events?

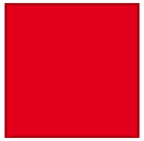

Component
x: int
y: int
width: int
height: int
onTouchEvent(eventType:int, f: function<void()>)
TouchEventListener
components: Vector<ComponentEventFunction>
add(eventtype: int, c: Component *, f: function<void()>): void
notify(eventtype: int, t: TouchData)
remove(eventtype: int, c: Component *)
GestureDetector
TouchData
struct ComponentEventFunction
c: Component *
onEventFunction: function<void()>

Event handling

Example

```
rectangle rect(10, 10, RED, 200, 200);

redLight.onTouchEvent(SINGLE_TAP, [this]() {
  setColor(GREEN);
  draw();
});
```

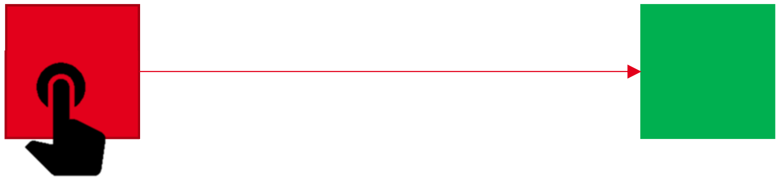


Event handling

Example

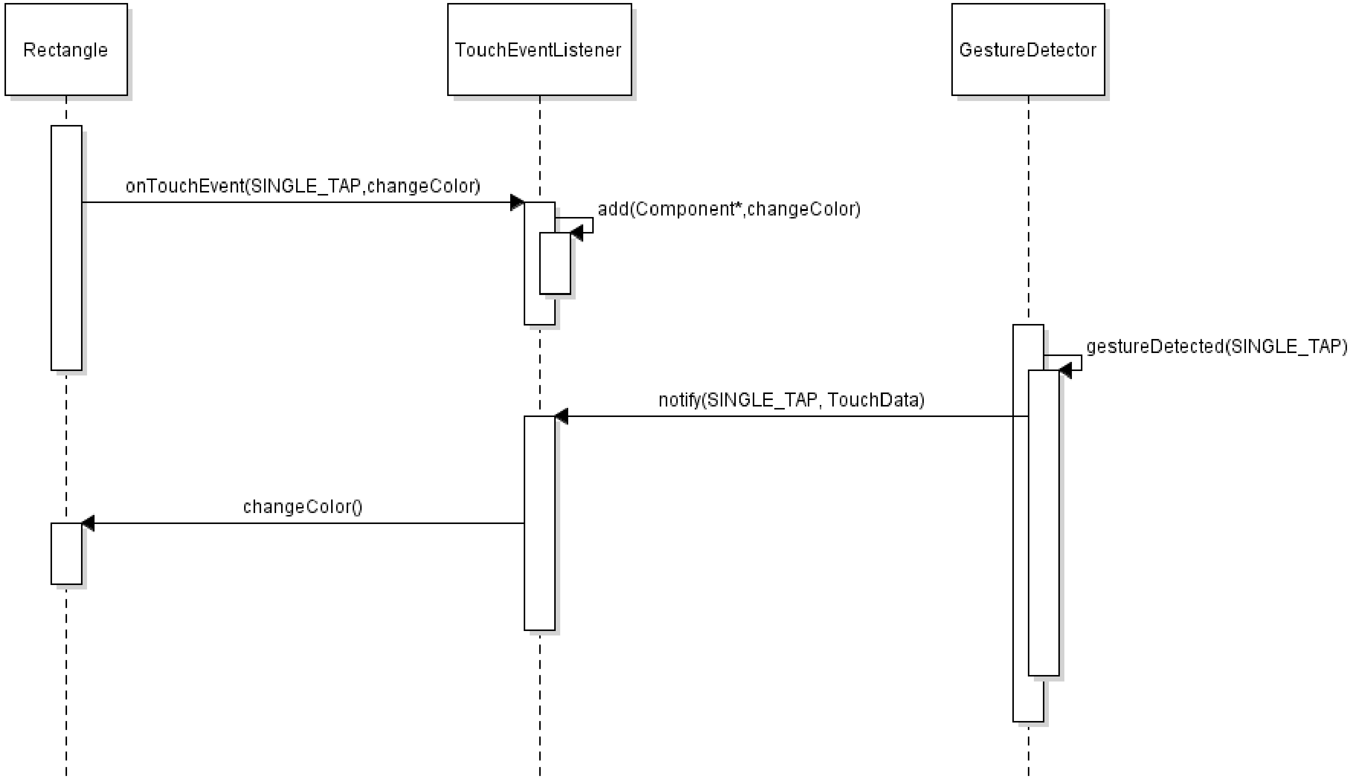
```
rectangle rect(10, 10, RED, 200, 200);

redLight.onTouchEvent(SINGLE_TAP, [this]() {
  setColor(GREEN);
  draw();
});
```



Event handling

How does the rectangle change its color?

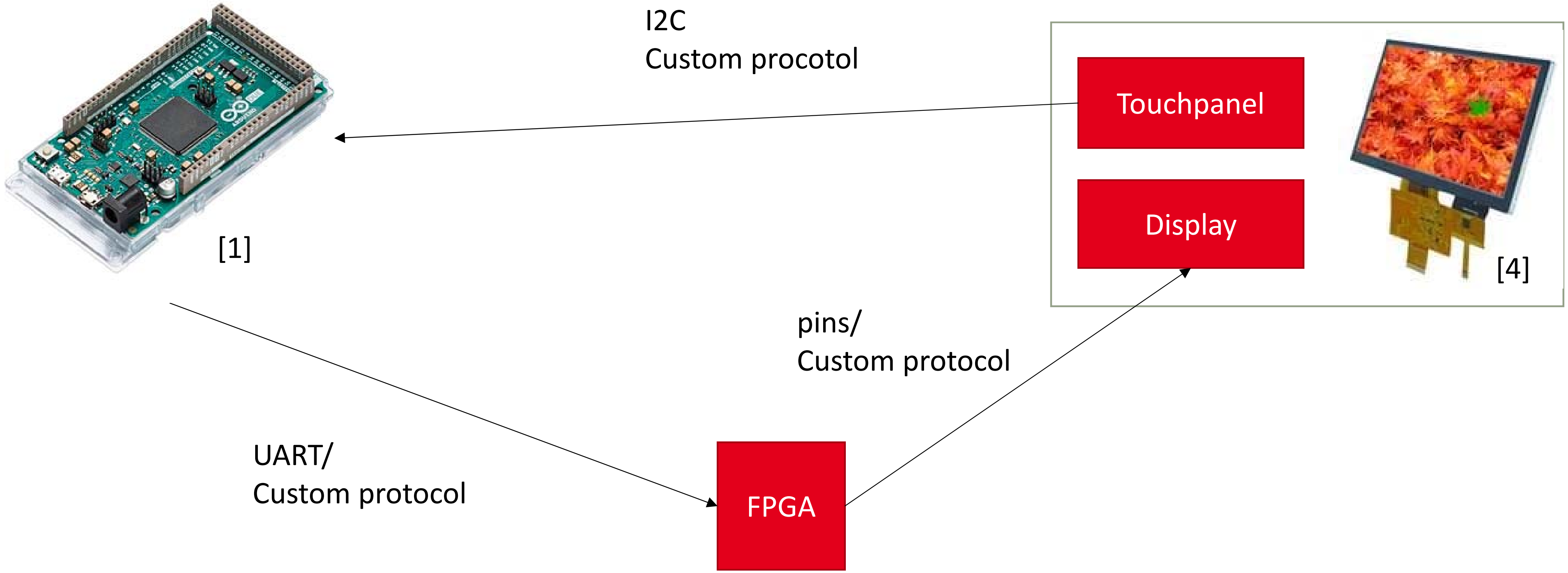

Rectangle
TouchEventListener
GestureDetector
onTouchEvent(SINGLE_TAP,changeColor)
add(Component*,changeColor)
gestureDetected(SINGLE_TAP)
notify(SINGLE_TAP, TouchData)
changeColor()

Software

Simulation

Software – Simulation

Why do we need a simulation?



I2C
Custom procotol

Touchpanel

Display

[1]

[4]

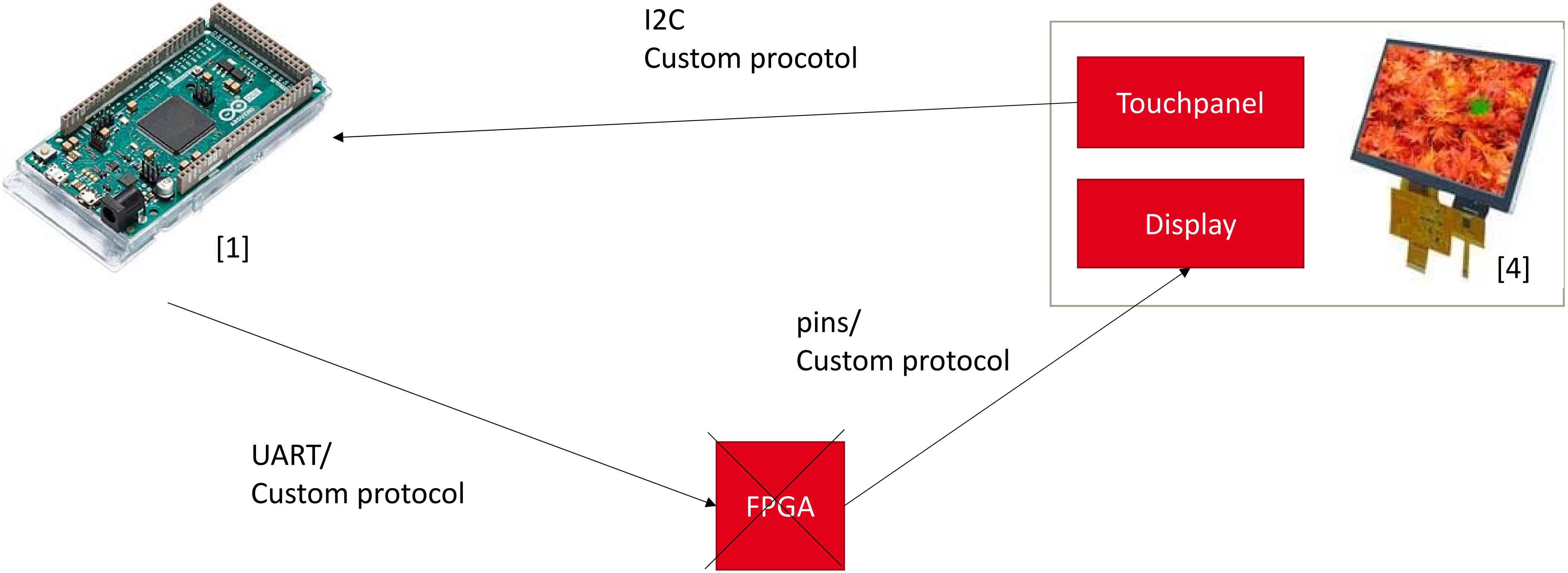
pins/
Custom protocol

UART/
Custom protocol

FPGA

Software – Simulation

Why do we need a simulation?



I2C
Custom procotol

Touchpanel

Display

[4]

[1]

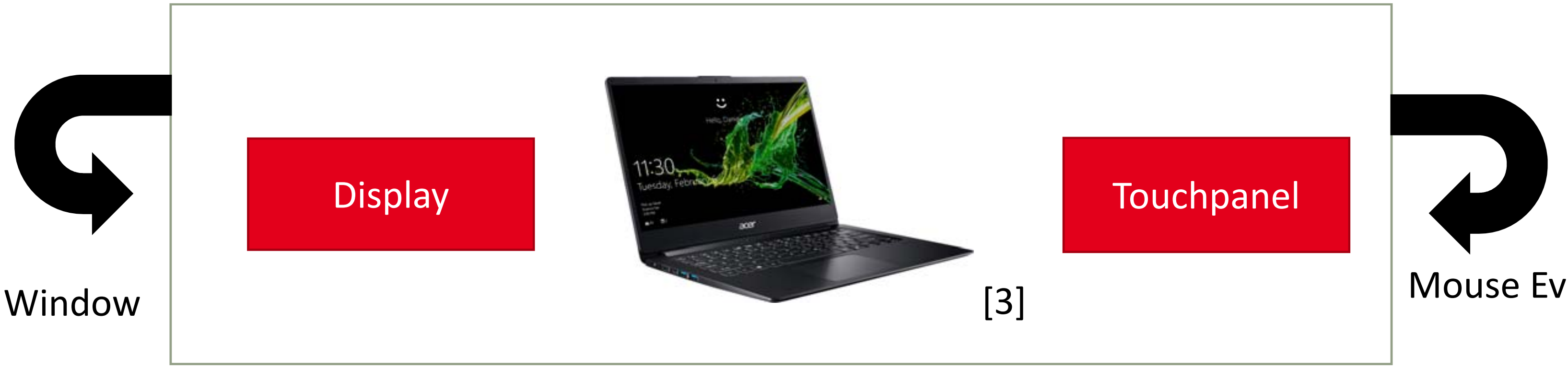
pins/
Custom protocol

UART/
Custom protocol

~~FPGA~~

Software - Simulation

Using a Windows PC to simulate Touchscreen and Arduino



Display

Touchpanel

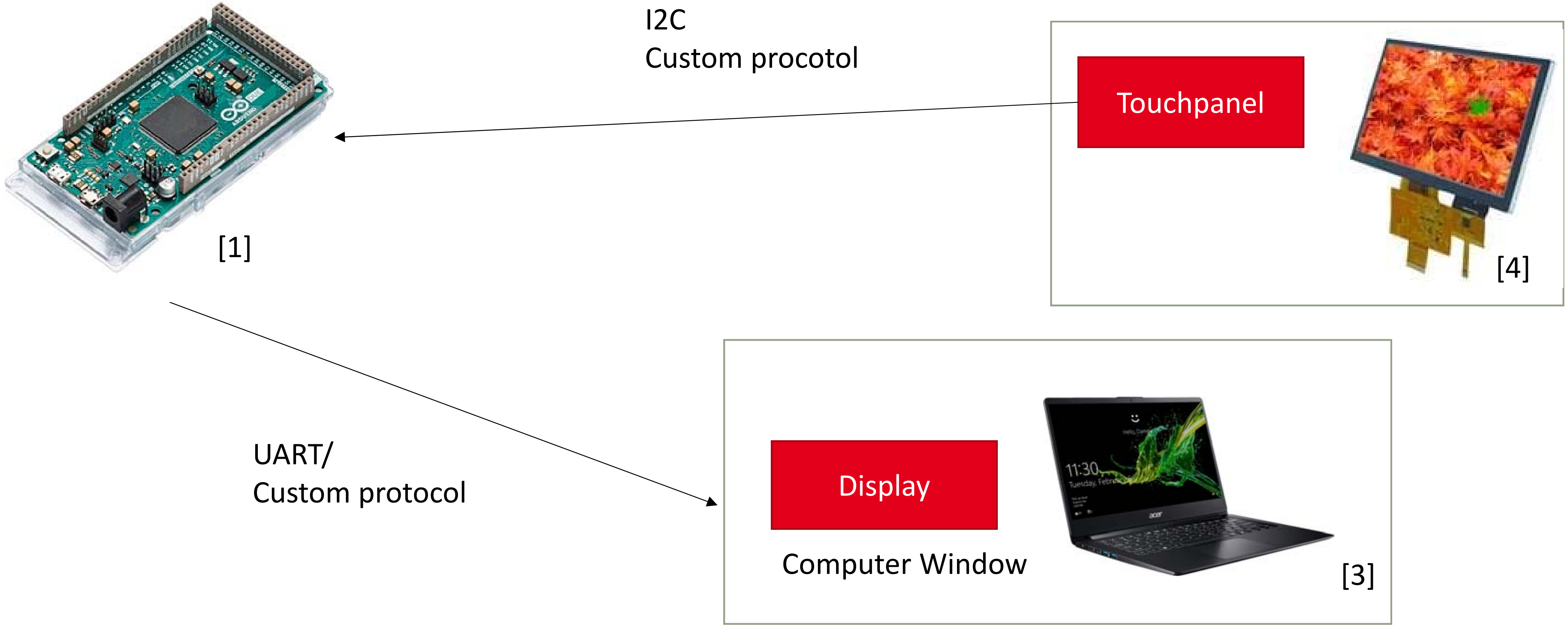
Computer Window

Mouse Events

[3]

Software – Simulation

Using a Windows PC as Display



[1]

I2C
Custom procotol

Touchpanel

[4]

UART/
Custom protocol

Display

Computer Window

[3]

Evaluation

Research Question

How am I answering these questions now?

- Research setup
  - Develop GUI library for given Touchscreen Display and Arduino Due
- Purpose
  - Using the Display during ES exercises
  - Investigate performance of setup
  - Explore use cases
- Goal
  - Detect Gestures reliably
  - Draw GUI Elements on Display
  - Maximize Quality of speed and memory usage
  - Derive recommendations for using the GUI library

Research Question

How am I answering these questions now?

- Research setup
  - Implemented Gesture Detection
  - Created drawable GUI Elements
  - Handled Touchevents
  - Simulated FPGA functions on Windows PC

Research Question

How am I answering these questions now?

- Goal
  - The System can detect Gestures reliably
    - Single tap, double tap, click and hold, dragging, …
  - GUI Elements on Display were implemented
    - Basic shapes, Buttons, Option groups, radio Buttons, …
  - Maximize Quality of speed and memory usage
    - Use bitmaps for faster drawing (8-times faster)
    - Sparsely use of Component attributes
    - Library: 4% ROM storage
    - Use fastest possible serial interface of Arduino (SerialUSB)

Research Question

How am I answering these questions now?

- Goal
  - Derive recommendations for using the GUI library
    - Touchscreens are very versatile and can adapt to many use cases
    - Quality of performance and memory usage are sufficient for applications without real time requirements
    - Should not be used for images/ streaming/ animations
    - Certain GUI Elements are more useful than others for Arduino (Textfields need Keyboard,…)
    - Use multiple pages instead of scrolling
    - Avoid large GUI elements
    - Avoid dynamic creation of GUI elements

Research Question

How am I answering these questions now?

- Purpose
    - Using the Display during ES exercises
        - Intuitive (OO) library
        - Easy to use for unexperienced students (Java inspired)
    - Investigate performance of setup
        - I identified the fastest way to send Data to display
        - Benchmarking by using Timers for functions
    - Explore use cases
        - Implemented Basic GUI Elements to be used for developing more complex GUIs
        - I could give recommendations when to use the library

Outlook

- Next steps
  - Documentation, testing
  - Multi touch recognition
  - Further Evaluation: Benchmarking, RAM usage

- Possible Extensions to the Project
  - GUI
    - Layout Manager, multiple pages,…
    - Anti aliasing
  - FPGA
    - Connect to FPGA
    - More complex commands to FPGA

Questions?

Sources - Images

- [1] Arduino: https://cdn-reichelt.de/bilder/web/artikel_ws/A300/ARDUINO_DUE_01_NEU.jpg
- [2] FPGA: https://www.intel.com/content/www/us/en/programmable/b/de0-nano-dev-board.html
- [3] Acer Swift 1: https://www.preis.de/produkte/Acer-Swift-1-SF114-32/5014185.html
- [4] Touchscreen: https://www.mouser.de/ProductDetail/ELECTRONIC-ASSEMBLY/EA-TFT070-84ATS?qs=f9yNj16SXrLEoCqnHqqifg%3D%3D

Sources

- Touchscreen: https://www.reichelt.de/tft-display-7-0-154x86mm-800x480-dot-24bit-rgb-pcap-touch-ea-tft070-84ats-p260435.html?CCOUNTRY=445&LANGUAGE=de
- Arduino: https://store.arduino.cc/arduino-due
- FPGA: https://www.digikey.de/de/product-highlight/t/terasic-tech/de0-nano-soc