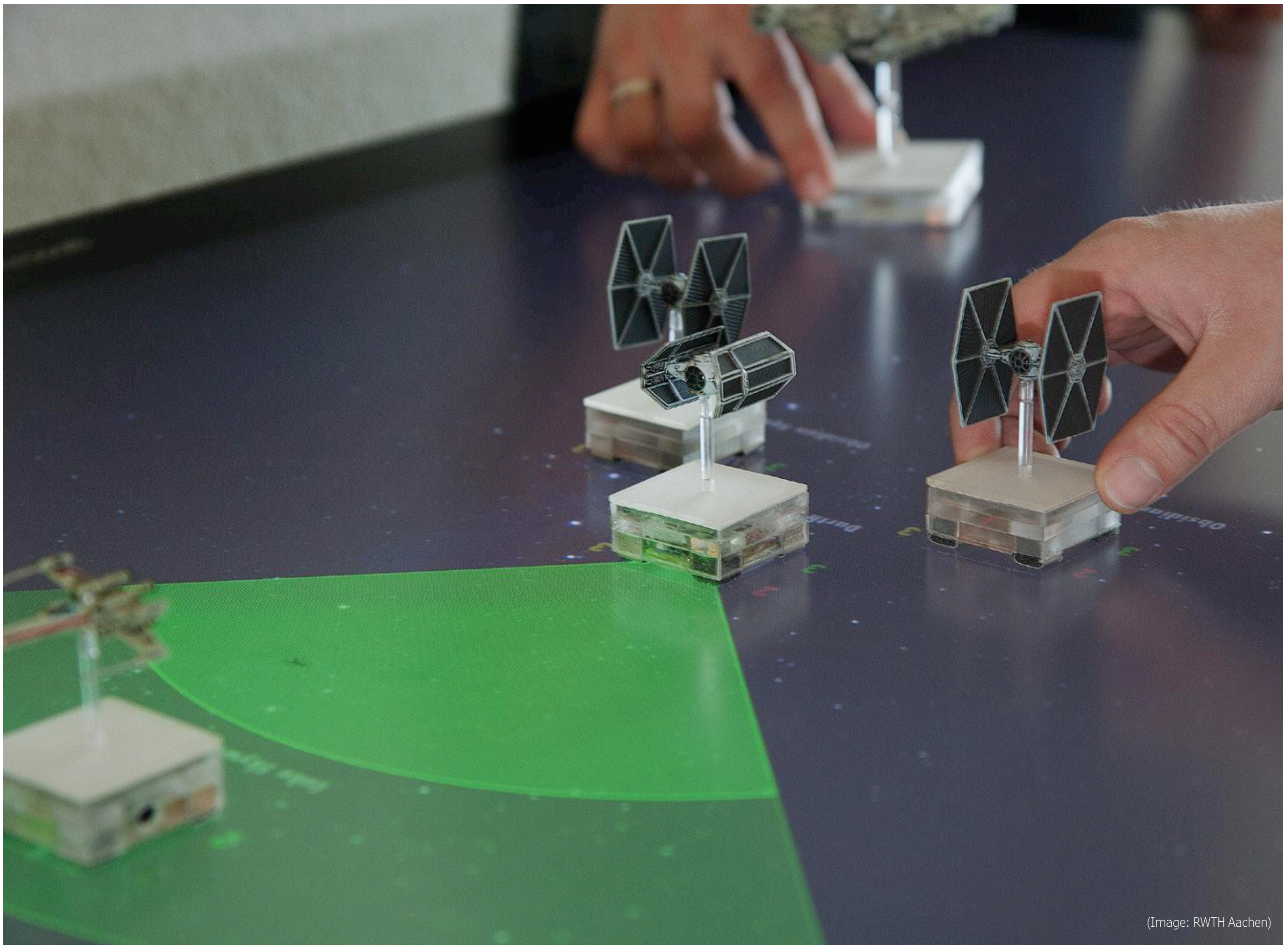
The TABULA Project

Capacitive detection of objects on a touchscreen



(Image: RWTH Aachen)

By **Christian Cherek**, RWTH Aachen, Germany

Large touchscreens, on which touchable objects, or 'tangibles', can be placed and moved around, are especially valuable in education applications. However, it has so far not been possible to use these in conjunction with capacitive touch sensors of the sort found on modern tablets and large touch panels such as the Microsoft Surface Hub. A research project at RWTH Aachen in Germany seeks to change this state of affairs, with a bit of help from Elektor.

In many places there is a severe shortage of IT professionals, and this applies not just in the IT sector itself: these days practically all businesses need IT skills. Training in such skills needs to be improved to enable more people to have a career in IT. In Germany, the Federal Ministry for Education and Research is supporting the TABULA project [1], whose goal is to make educational materials more intuitive and less dry, dusty and abstract.

The idea behind the project is that students can move physical objects around on a (horizontal) touch-enabled display which in turn presents interactive educational content, both in the area around the object and between objects. The control computer must therefore be able to identify each of these 'tangibles' and their position at any point in time. If it is desired to use a modern tablet or, even better, a large touch display that uses capacitive touch sensing, then there are several challenges to be faced.

The project and the people

The project is supported by the **German Federal Ministry for Education and Research** (BMBF). Two demonstrators will be developed in the course of its three-year duration. The first will illustrate data structures and algorithms, while the second has a focus on flow-based programming.

Bundesministerium für Bildung und Forschung

Three groups at **RWTH Aachen** are involved in the project.

- The Media Computing Group, led by Prof Jan Borchers, has been investigating modern forms of interaction for many years. Researchers there are among the first in the world to have succeeded in developing tangibles that work with multi-touch screens and that can be detected even when the user is not touching them. The department will work with Elektor to develop the tangible technology. Meanwhile, the Learning Technologies research and teaching group, under Prof Ulrik Schroeder, will bring to the project its know-how and experience in interactive learning games and in imparting IT skills. Finally, Prof Martina Ziefle's Communication Science group will look into the ethical and social implications of the technologies. See http://hci.rwth-aachen.de/TABULA for more information.

- **Elektor** will work with the Media Computing Group at Aachen to improve the tangible technology, and will report on developments in the magazine. It will also offer tangibles for sale to interested parties after the end of the project.

- **inside Media AG**, whose authoring tool has won many prizes in recent years, will bring its know-how in creating content for e-learning environments to the project.

Logos: public domain

The problem

The user can place the tangibles on the display and move them around. Applications that can make use of this range from music production and computer science education, for example illustrating search trees and graphs, to medical simulation for training. The possibilities for the feedback that such a system can provide to this form of user input enable a new kind of interaction. Similar systems do indeed already exist, generally relying on analyzing reflections of infrared light to determine the position of the tangibles and detect when the screen is touched. The reliability of such systems is strongly influenced by ambient lighting conditions, and direct sunlight in particular can be a significant problem. For this reason the trend is away from optical systems and towards capacitive systems, which are less susceptible to the external environment. Capacitive multi-touch screens, which can recognize more than one contact at a time, contain transmitter electrodes which are enabled rapidly in sequence one after another, and, spatially separated from them and arranged perpendicularly to them, receiver electrodes (see **Figure 1**). An electromagnetic field is therefore set up between the electrodes at each intersection. When a conductive object such as a human finger comes near, the field is affected. The touch controller detects this change and reports the coordinates of the point where the touch occurred. Tangibles for use with capacitive multi-touch screens are usually enclosed in an electrically conductive material, or may be operated using a conductive stylus. The electrical connection between the top of the object and its base in effect allows the capacitance of the user touching the object to be transmitted through to the conductive base of the tangible, and thence to the upper surface of the touch-sensitive display, to a sufficient extent to allow the object to be detected. In order to differentiate between different tangibles, the base of each is designed with a different pattern of touch points that can be recognized in software.

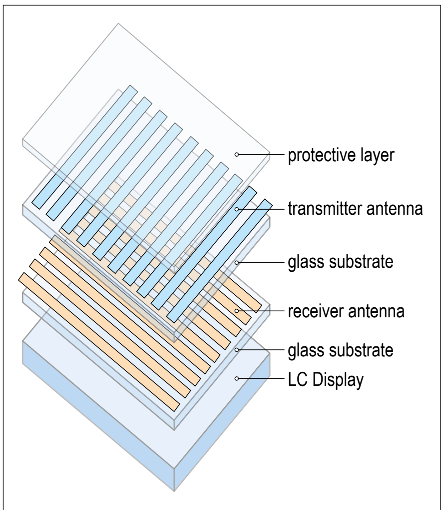


Figure 1. Construction of a capacitive multi-touch screen.

At first sight this approach seems to present few difficulties. However, getting into the details we discover that there are few challenges to be overcome (not that we don't like challenge!). Even a conductive object

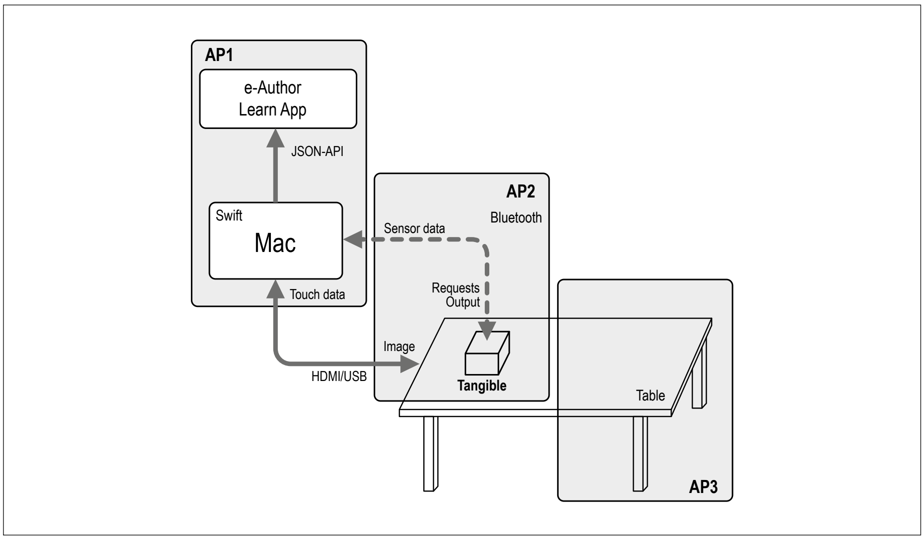


Figure 2. The TABULA system consists of three parts: control computer, tangibles and touchscreen.

only presents a very tiny capacitance change at the point of touch. Two disadvantages flow from this:

- Tangibles can only be detected on the capacitive multi-touch surface when the user touches them. When the user lets go of the object the capacitance falls back and the system cannot determine whether the user has simply let go of the object or has removed it from the screen.
- Discriminating between tangibles purely on the basis of the characteristic pattern of touch points on their bases is only possible for a small number of tangibles simultaneously.

Both these problems have already been solved in the preliminary planning of the TABULA project. Before going into the nitty-gritty, let us take a look at **Figure 2**, which shows the basic architecture of the system in terms of hardware and software. There are three components to the project. The touchscreen (AP3) sends touch data over USB to a control computer (AP1) and simultaneously receives from it image information, which will depend on the positions of the tangibles. Running on top of this is the application software, educational or otherwise, which processes the raw data into feedback accessible to the user to reflect the logic demanded by the application in question. The tangibles (AP2) on the touch surface were initially conceived of as purely passive, but in the course of the project it has become apparent that they must instead be active devices. They are therefore designed to communicate with the control computer over a Bluetooth connection, transmitting among other things their position and identity. So now let us look in more detail at how the project has evolved.

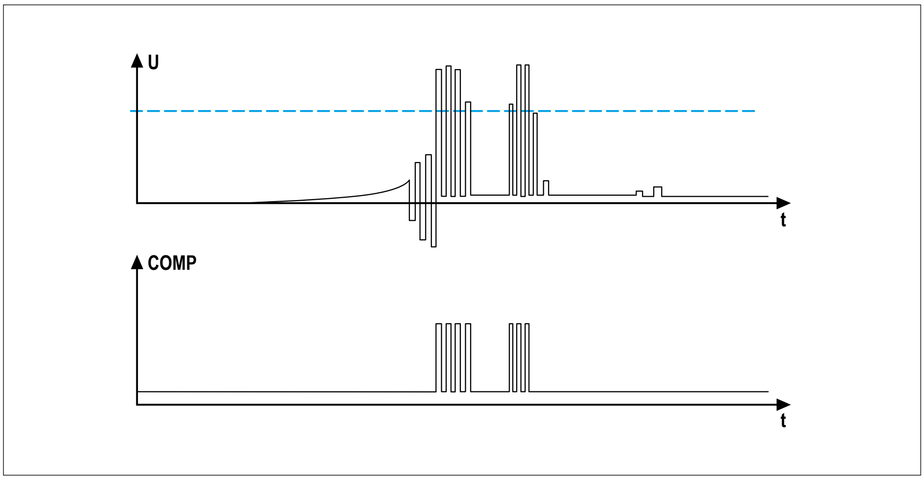


Figure 3. Above, the signal emitted by the capacitive multi-touch screen when looking for touch points; below, the output of the comparator as seen at the input to the microcontroller.

Recognition

In 2013 the Media Computing Group at RWTH (Technical University of Rhine-Westphalia) Aachen found a way to solve the first of the above problems. If the tangible is fitted with several conductive pads and these are electrically connected together, for example using copper tape, then the location of the object can be pinned down even when the user is not touching it.

The trick here is that when a transmitter electrode is driven under one of the pads of the object, the other pads will be over inactive electrodes on the touch surface. The capacitance to ground of the touchscreen itself will be sufficiently high that the connection to the pad that is over an electrode that is not being driven will create enough total capacitance to allow the first pad to be detected. This principle is similar to that of a stylus except that in this case it is not the capacitance of the user that is being exploited, but rather that of the table itself. If the pads are arranged in a pattern of three points in a right-angled triangle then the tangible can always be detected.

However, this solution creates a new problem. Modern capacitive systems have powerful filtering mechanisms to improve immunity to external interference from the environment. One effect of this filter is that a touch point that does not move (such as our tangible that is not being touched by the user) will after a while be filtered out. So now we have the problem that the system cannot distinguish between a tangible that has been filtered out and one that is not present.

Two years later, in 2015, the RWTH team found a solution to this in the form of an active tangible. This new design of tangible is equipped with a small antenna and a comparator that can pick up the signal from the touch surface and hence determine when the tangible is in contact with it. **Figure 3** shows a typical signal that might be measured directly on the surface of a capacitive multi-touch screen. The signal is repeated at regular intervals (usually at 60 Hz) to detect the position of a touch. The simple antenna in the tangible picks up the signal and filters out any parts that fall below the threshold line shown, resulting in the signal at the bottom of the figure.

The voltage level corresponding to the threshold line is adjusted using a potentiometer connected to the second input to the comparator: the circuit is shown in **Figure 4**. It is essential to be able to adjust this level, as the strength of the signal that can be picked up varies between different models of touch panel.

The tangibles are also fitted with a photodiode as a back-up: see **Figure 5**. This is used to detect light from the screen. Here again the sensitivity of the circuit can be trimmed using a potentiometer to adjust the brightness threshold.

The information so gathered is sent to the control computer over Bluetooth, and is enough to enable it to distinguish between a tangible whose presence has been filtered out by the touch panel and one which has genuinely been removed from the panel. One happy side-effect is that the Bluetooth chip in the tangible has its own UUID code, which makes it easy to distinguish between several different tangibles even if they have the same pattern of conductive pads, and this solves the second of the two problems mentioned above.

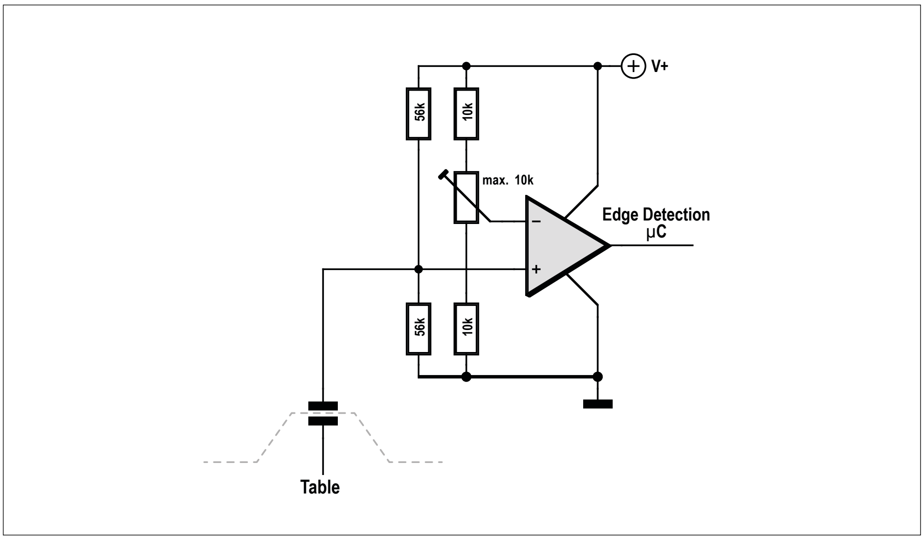


Figure 4. Circuit for signal detection in the tangible. The potentiometer allows the threshold voltage to be set.

Software

On the software side the system is composed of two parts: the software running on the control computer driving the multi-touch table, and the firmware running in the microcontroller in the tangible. In order to detect possible locations for tangibles the control computer software starts from the data received directly from the table's touch sensor. If the table reports a touch point then the software looks to see if it can find any of the patterns of interconnected pads it has been trained to recognize.

In our implementation the control computer is an iMac. The software, based on the 'Multitouch Kit' Swift framework, receives raw data from the touch panel driver and then processes it to find and discriminate between touch points and tangibles. Since each tangible has a fixed pad pattern, the framework can use this to determine both the location and orientation of each tangible.

The job of the firmware running on the tangibles is to help avoid the control software confusing one tangible with another, or with fingers touching the screen. To keep the firmware as simple and flexible as possible, an ATmega85 microcontroller is used in conjunction with an HM-11 BLE (Bluetooth Low Energy) module. This minimizes the energy required to communicate with the control computer. As a result a tangible fitted with three fully-charged AAA cells should be able to run for around 24 hours.

Comparator

When a tangible comes into close proximity to the touch panel, the firmware detects that the comparator output has been triggered and sends an '`ON-table`' event to the control computer. The tangible uses a timer to check whether the signal is received at regular intervals from the table: **Figure 6** illustrates the routine responsible for this. Whenever an edge is detected on the comparator output the timer is reset. The Multitouch Kit code can now organize the detected touch points into individual tangibles. Even if a tangible remains in the same position for a long time and its touch points are filtered out by the panel, the software running on the control computer will nevertheless know from the sensor data that it receives that the tangible is still sitting on the table. Then, when the tangible is taken off the table, the 40 ms timer will expire because the tangible is no longer picking up the signal from the table. This causes the microcontroller to send an '`OFF-table`' event, and the control computer will then remove the tangible from its list of objects known to be on the table.

The display driver reports touch events as a path of motion rather than as a single point, and so a little processing is required to extract the location of each tangible. A moving tangible is usually rather easier to locate as the hardware detects moving touch points better than stationary ones. Any touch points that the software cannot account for as part of a tangible is interpreted as a finger touch, and

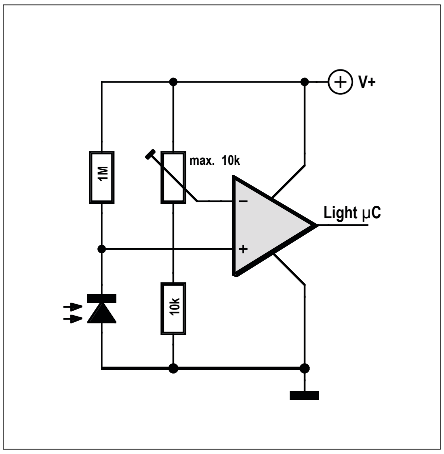


Figure 5. Circuit for the light sensor in the tangible. Again, a potentiometer is provided to allow adjustment of the threshold voltage.

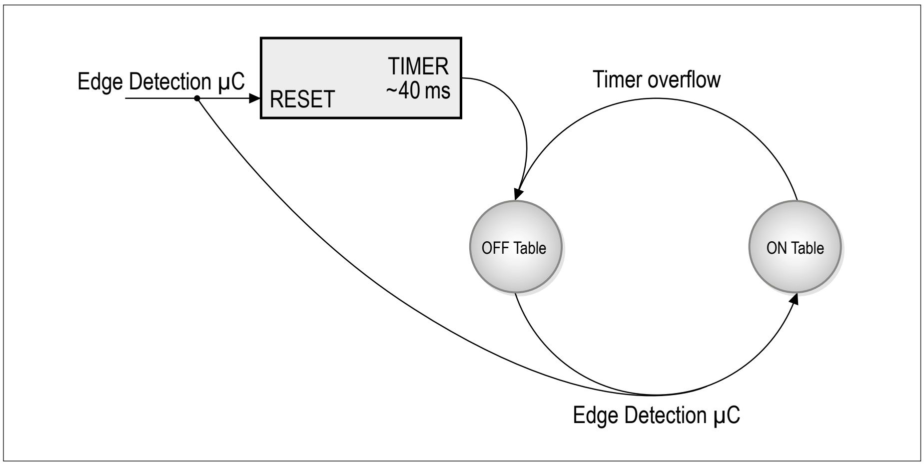


Figure 6. In this state machine the comparator output is used to reset the 40 ms timer whenever a signal is detected, and triggers an 'ON-table' event.

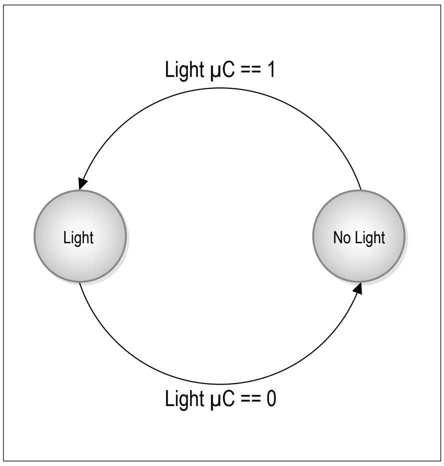


Figure 7. The state machine for the light sensor. It takes a very bright spot of light on the screen to trigger an event.

Table 1. Protocol tree structure used by the tangibles to send status information to the control computer.

```
controller tree protocol
 controller
 |
 +- [0: // advanced communication port
 |
 +- [1: // product identification
 | +- 0: // manufacturer (like: rwth_i10)
 | +- 1: // product id (like: 0A4F)
 | +- 2: // uid (like: 0FA457EF)
 |
 +- [2: // basic attributes
 | +- 0: // on table?
 | | +- 0] // off table
 | | +- 1] // on table (with side 1)
 | | +- 2] // on table (with side 2)
 | | +- .. // and so on..
 | |
 | +- 1: // see light? (the side that is on the table)
 | +- 0] // see no light
 | +- 1] // see light
 |
 +- [3: // orientation, position
 | +- 0:FFFFFFFFF] // orientation (single rotation vector xxxyyyzzz)
 | +- 1:FFF] // orientation relative to table
 (| +- 2:FFFFFFFFF] // position)
 |

Examples:
 [2:0:1] // placed on the touch table
 [2:0:0] // taken off the touch table
 [2:1:1] // sees the licht
 [2:1:0] // sees no light
```

indeed it is possible to detect multiple manual touches simultaneously alongside tangibles.
All these measurements and tests are done in real time in parallel with the processing done by the touch panel, and so the communication between the various parts of the system is practically transparent to the user.

Light sensor

It is possible that the user might accidentally make the same touch pattern with his fingers as that produced by a tangible, or that two tangibles are placed on the table simultaneously and the control software cannot determine which Bluetooth signal corresponds to which tangible. In such cases, and more generally when the system wants to verify the position of multiple tangibles on the table, the control software will produce tiny, very bright, spots of light at the positions where it thinks tangibles might be located. When a tangible detects such a spot of light under it, it sends a 'LIGHT' event to the control computer. The system can use this information to correctly match up each tangible with its touch points.
The light sensor can only work reliably when detecting significant changes in brightness. The tangible is therefore designed to send the 'LIGHT' event only when the display is very bright. The sensor is not really suitable for making a more subtle measurement than just a binary decision between 'light' or 'no light' (see **Figure 7**). However, this is perfectly acceptable for our purposes.

Protocol

The protocol which the tangible uses to send its messages to the control computer is designed to make it as easy as possible to extend it with new functions. The basic firmware running on the tangible, which we describe here and which is contained in the package available for download, simply sends events from the two sensors to the control software. Each message that the tangible can send corresponds to one leaf of the protocol tree illustrated in **Table 1**. So, for example, if message 2:0:1 is sent then it means that the tangible has been placed on the table. So far the only message types implemented are the 'ON-table' and 'OFF-table' status messages and 'LIGHT' and 'NOLIGHT', but it is easy to extend this within the tree structure. For example, a tangible in the form of a die might be able to send a range of different 'ON-table' events depending on which of its faces is touching the table, and the corresponding messages might be 2:0:2, 2:0:3 and so on.

We have used the 'Microsoft Surface Hub 55' and 'Microsoft Surface Hub 84' touch displays, both of which are capable of recognizing up to 100 touch points simultaneously. This gives plenty of scope to use a large number of tangibles with capacity left over for detecting finger touches. The Multitouch Kit also runs under iOS, and so the tangibles can also be used on iPhones and iPads.

Both systems allow application development using Objective C or Swift, for example using Apple's APIs for graphical applications (SceneKit or SpriteKit). It is possible to associate a tangible with a digital object and in turn associate the tangible with a graphical object on the screen. The graphical object will then follow the movements of the tangible automatically. There is the further advantage that one can then easily make use of the engines supplied as part of Apple's code to produce a range of effects, and, since the updates included as part of iOS 11, even add artificial intelligence functionality, all without having to write any extra code oneself. The required association is made directly within Apple's programming environment and all that remains to be done is to write a native application that supports the tangibles. The tangibles can of course also be used in conjunction with any other programming language that includes functions for processing multi-touch events. It is even possible to develop applications rapidly in HTML5 that connect a virtual object to a tangible.

What next?

Summer 2017 marks the end of the first year of cooperation of the TABULA project partners, and two years now remain. Elektor is supporting RWTH Aachen in the development of hardware and firmware for the tangibles. The complete current version of the source code for the microcontroller firmware is available for download from the project pages accompanying this article [2]. The main part of this code comprises the two state machines and the implementation of the protocol the tangible uses to send data to the control computer.

The circuit diagram and printed circuit board layout for the first prototypes was completed just as this article was being prepared. **Figure 8** shows the first 'production-ready' tangible hardware. The complete schematic, the board layout and the parts list are all included in the download at [2].

(160123)

Web Links

[1] http://hci.rwth-aachen.de/TABULA

[2] www.elektormagazine.com/160123

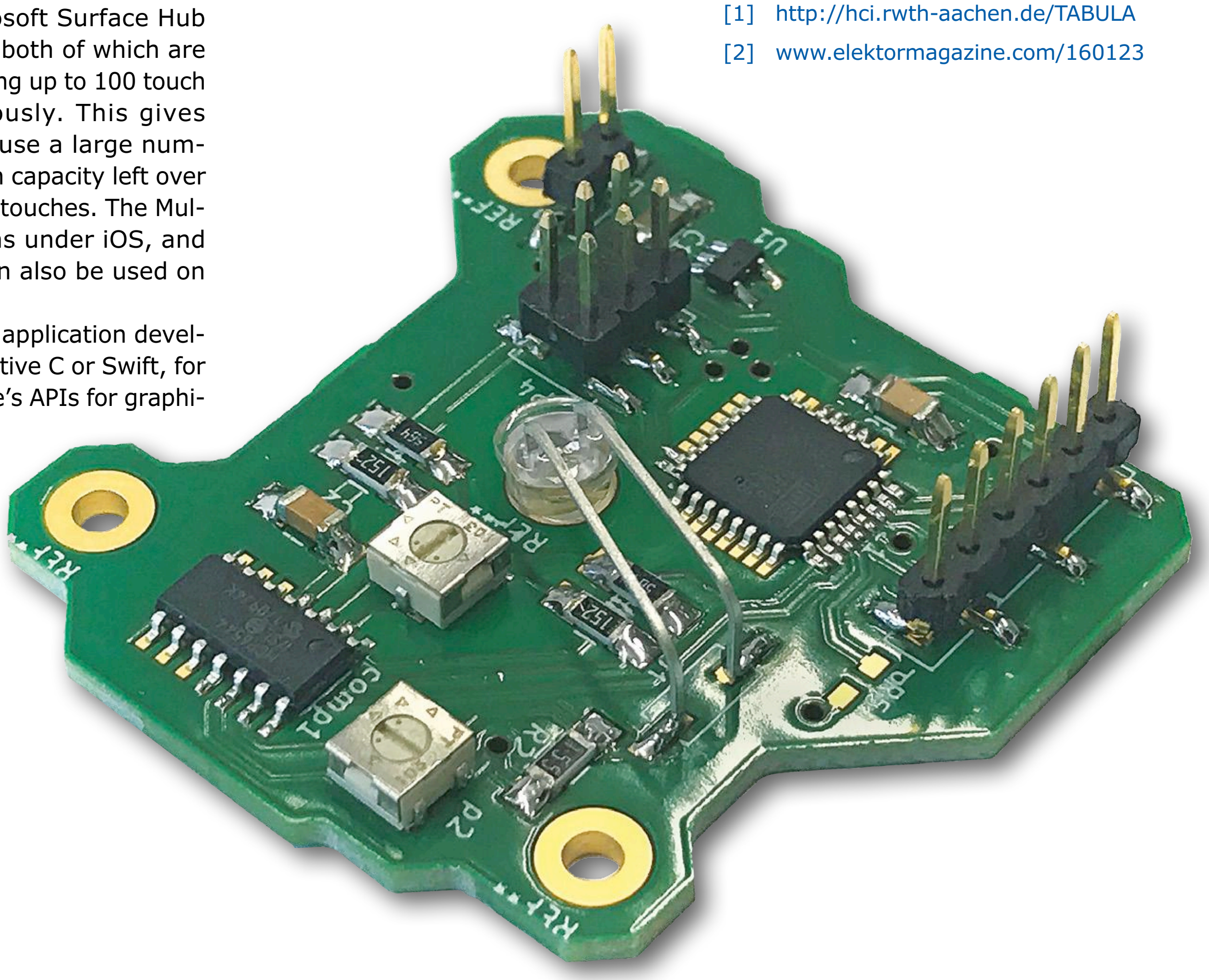


Figure 8. The populated printed circuit board with opamps, trimmer potentiometers and microcontroller. The Bluetooth module is not fitted. The light sensor in the middle looks downwards onto the touchscreen. (Image: RWTH Aachen).