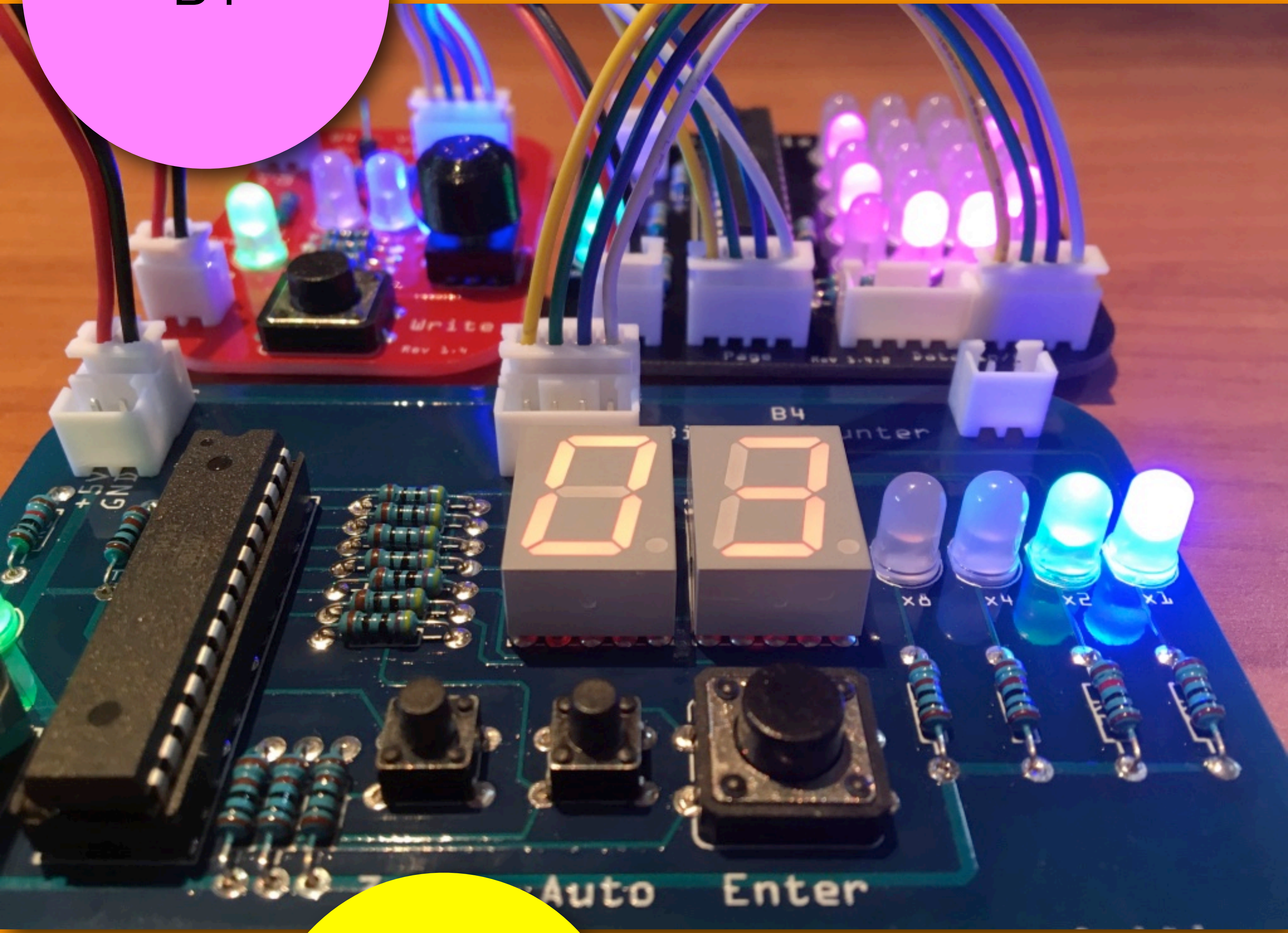
B4



The comprehensive, modern, experimentation-based course unlocks the fascinating world of Digital Technologies.

Primary School Starter Kit

0010
0000
0010
0000

DIGITAL
TECHNOLOGIES
INSTITUTE

“Those who can imagine anything, can create the impossible.”

(Alan Turing)

Table of Contents

Safety instructions

⚠

WARNING:
CHOKING HAZARD - Small Parts
Not for children under 3 years.
PHOTOSENSITIVE EPILEPSY -
Some of the experiments produce light flashes that can potentially trigger seizures in people with photosensitive epilepsy

The B4 operates on 5 Volts and only draws a few milliamperes. Nevertheless, it is an electrical device and should be handled as such. We recommend to treat it with care and to keep it on a non-conductive, dry and level surface. Do not scratch the surface of the printed circuit boards with sharp or metallic instruments, as this might damage the wires.

Acknowledgements

We would like to thank Charles Petzold, the author of ‘Code: The Hidden Language of Computer Hardware and Software’, published in 1999. His book has both inspired and guided the design of the B4. We recommend it as additional reading material for students.

We would further like thank Henrik Maier from proconX for his guidance and feedback on the electrical engineering design, fabrication and component selection, which has been invaluable to transforming the B4 from a breadboard prototype to a robust design that can be used in the classroom.

Special thanks to Dr. (med) Hayden White for his support and input which have been invaluable to get the B4 off the ground. His regular feedback on the development of the B4 has influenced many of the design decisions.

A big thank you is owed to Martin Levins, Katie Woolston and Michael Schulz for their contributions to this handbook. Esther Schulz contributed to the design of the characters and symbols of the Dot Matrix Display. David Schulz has contributed code that drives the seven segment LEDs of the Binary Counter and the Decimal Display. He initially developed this for his Young ICT Explorers project, The Tardis, in 2016. We’d further like to thank the Arduino community: Three of the B4’s modules deploy an Atmega processor which run Arduino programs. Keep up the great work!

Mrs. Sharon Singh and her year 8 students at St. John’s Anglican College in Forest Lake, QLD, have provided great input on the B4 and this handbook, which has lead to many improvements and new experiments. Thank you!

Dr. Karsten Schulz, CEO, The Digital Technologies Institute.

Included Parts

1x Adder
1x Binary Counter
1x Decimal Display
1x Dot Matrix Display
2x Variable

5 x 4 Pin Wires
5 x 2 Pin Wires
1x USB Cable

1x Printed Student Handbook

Power Consumption:
5V, 200mA (average), 1W DC.

This product complies with the Restriction of Hazardous Substances Directive and is lead free.

The illustrations in this handbook can slightly differ from the actual modules. However, the functionality is the same.

This handbook has been made with great care. Should you find errors or have ideas to improve it, please email us at enquiries@digital-technologies.institute.

Designed and manufactured in Australia

Hello Parents and Teachers

Digital Technologies play an increasing role in our lives. This Primary School Starter kit contains the most important components to conduct exciting and practical Digital Technologies experiments in the classroom and at home. We have made it to help students explore the fundamental operation of computers. It has been designed from the ground up to support students who are studying the new Australian Curriculum: Digital Technologies. The B4 supports the teaching of the knowledge and understanding of digital systems and the representation of data. In this course, we also cover binary addition, binary to decimal data conversion, and pixel graphics.

In particular, this course helps students to understand variables, which are a notoriously hard and abstract concept for the young minds before year 7. The B4 makes variables tangible. Or have you ever held a variable in your hands? The B4 makes this possible.

Digital Technologies areas covered: Digital systems, representation of data, collecting, managing and analysing data, investigating and defining, generating and designing, producing and implementing, evaluating, collaborating and managing.

The B4 makes all these accessible with a practical hands-on approach that does not require a laptop or tablet computer.

A virtual version of this hardware kit is available at https://mycomputerbrain.net/php/courses/psk.php Teacher and parent accounts are free and student accounts can be purchased at a reasonable price. We recommend to use the physical kit in conjunction with the virtual kit.

Hello Students

Many years ago, when our grandparents were young, computers were big - bigger than our bedrooms. Their parts were also quite big and you could hear and see them working, or computing as we say. Modern computers have become very tightly integrated and now fit into the pockets of our pants. But this means that their parts have become so small that we can hardly see them, and even less distinguish them with our own eyes. Therefore, the B4 has bigger components that you can easily see. Whereas the speed of modern computers is measured in millions of instructions per second the B4 operates at human speed, thus allowing us to see with our own eyes how data flows in a computer.

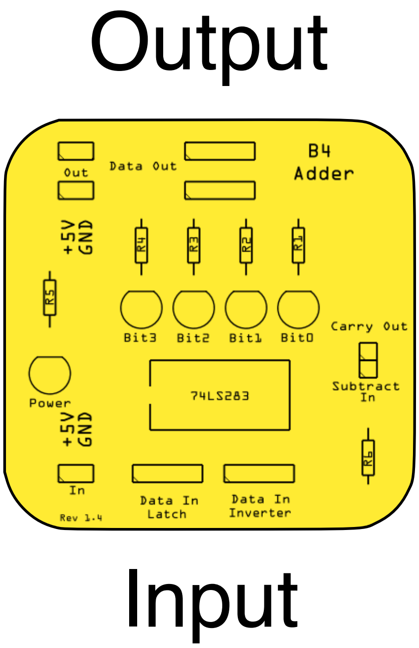
Together, we will do lots of exciting experiments in this course. Don’t be afraid to try things out.

B4’s Parts

The B4 Primary School Starter Kit contains the most important computer modules for primary students to get started in Digital Systems, Binary Data Representation, Variables, and Pixel Graphics.

The B4 Primary School Starter Kit consists of 6 computing modules that represent core functions of a computer, such as binary counting, variables, adding, and graphics. With these, students undertake experiments which familiarises them with the binary number system and how computers work inside.

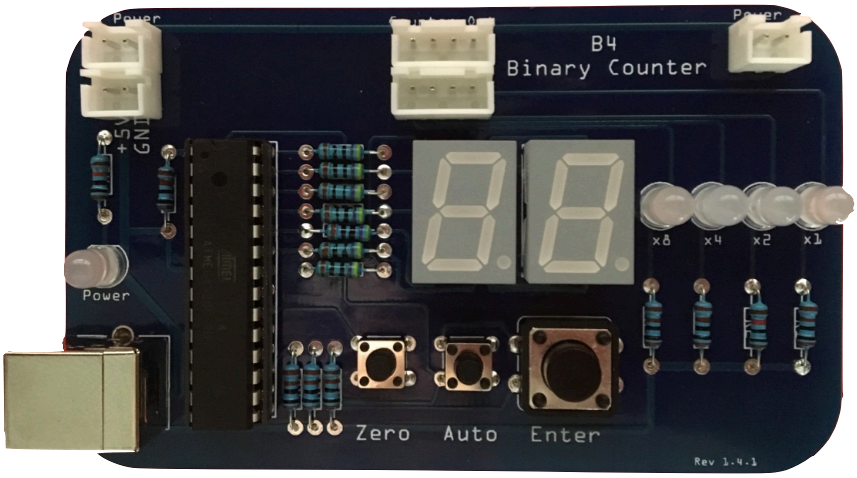
Generally, each module receives its input through the connectors at the lower end of the module and provides an output through the connectors at the top. All connectors are labelled with In or Out.



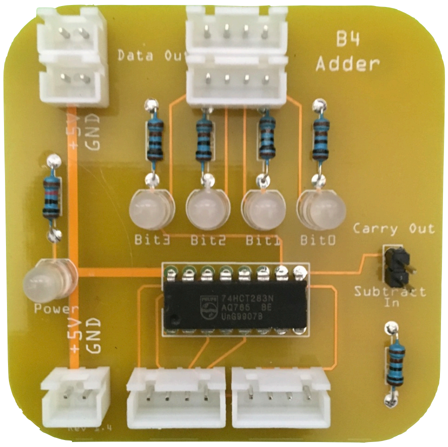
We can further classify the B4 modules into counting, mathematics, input and output. All modules within a category are of the same colour. The counting modules are blue and the mathematics module yellow. The input modules are red and the output modules are black.

Modules	Function	Color
Binary Counter	Counting	blue
Adder	Mathematics	yellow
Variable	Input	red
Decimal Display, Dot Matrix Display	Output	black

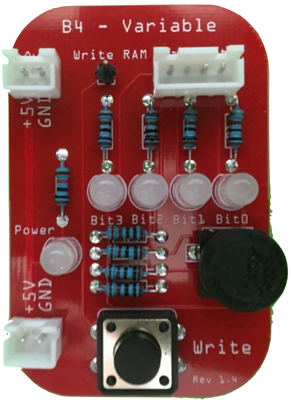
The modules are all labeled. Take them out of the box and find each of the modules as we describe them below. Let’s start:



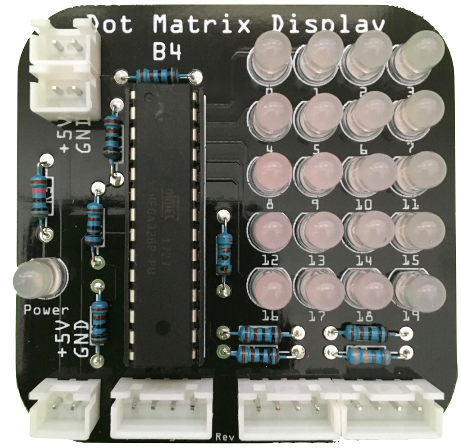
The **Binary Counter**’s main function is to count from 0 to 15. Every time you press the Enter button, the number on the display will grow by one. When it shows 15 and you press the button, the counter will tick over and start at 0 again. The Binary Counter also has a Zero button, which resets the output to 0 and an Auto button, which makes the Binary Counter work really fast.



The **Adder**, as the name suggests, is a small machine that adds two numbers.



With the **Variable** we can produce binary data simply by rotating the knob. The B4 ships with two Variable modules. The rotary knob of the Variable was made on a 3D printer.



The **Dot Matrix Display** has 20 pixels, which are arranged in a 4x5 matrix. At the bottom of the module is a page input port and two data input/output ports. With these we select which pattern we want the pixels to show, for example a letter, a number, a smiley, or something else. For a summary of all the patterns that the display can show, refer to Appendix A.

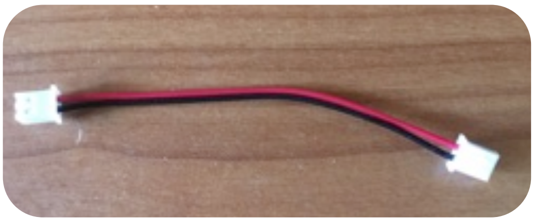
We now have a basic understanding of the modules of our B4. Don't worry if you haven't understood everything yet. We will revisit each module in more depth during the following experiments.

Wires and Connectors

In order to connect the B4 modules with the computer and with each other, the B4 comes with 4 types of wires. They are:

A **USB cable** to provide electricity from a power source to the B4's Binary Counter module and from there, to all other modules connected to the Binary Counter. You can connect the USB cable to a PC, Laptop, USB Hub, USB battery, or any other suitable 5V power source with a USB port.

2 pin **power wires** with black and red wires. They are part of the B4's power distribution system and transport electricity from module to module. Each module has one power input and 1-2 power outputs.



4 pin **data wires**. These transport 4 bit data between the modules.

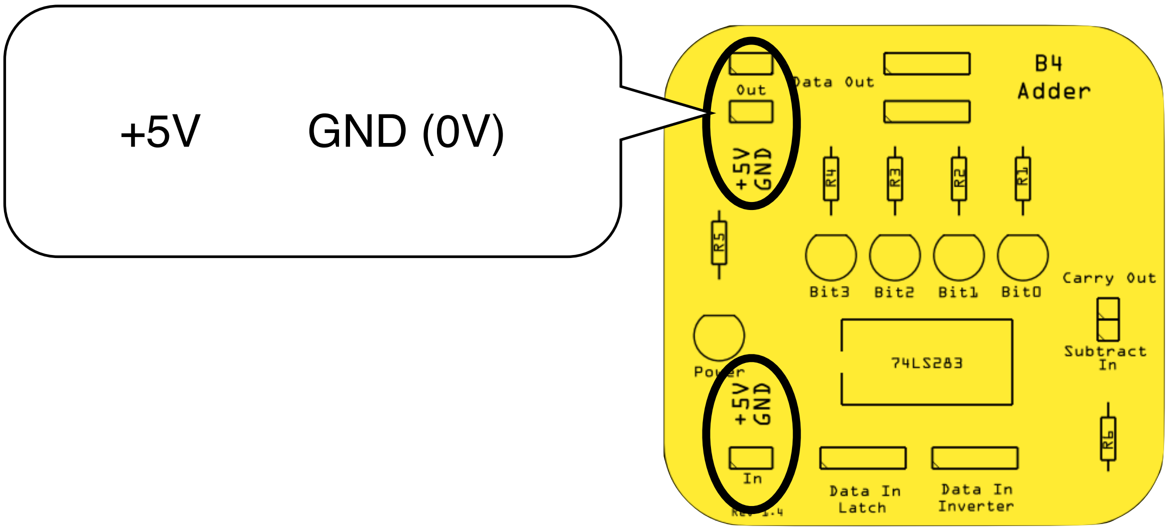
You find corresponding connectors on the modules. The 2 and 4 pin connectors are directional and the wires will easily click into them. Unless you apply excessive force you should not be able to accidentally plug them in the wrong way.

In the diagrams in this book, we use the following wiring notation. A thin solid line denotes a power wire. A thick line denotes a data wire. This is just to make the setup a little bit easier for you.

Symbol	Meaning
―	power wire
▬	data wire

Electricity

Each of the B4’s modules has a electric power distribution system on the left hand side of the modules. With the exception of the Binary Counter, which connects to a USB port, the other modules have power in and out connectors.



+5V is on the left and GND (Ground, or 0V) is on the right. The wires will always connect in the right way.

Please look after me

The B4 is fairly robust and will last a long time with proper care. As long as you don’t plug wires into connectors they are not designed to go in and as long as you don’t drop the modules, step on them or use them as a doorstopper, things should be just fine.
Always only plug the 2 pin wires into 2 pin connectors. the same applies to 4 pin wires and connectors. **Under no circumstances plug a 2 pin wire into a 4 pin connector**.

About Binary and decimal numbers

To distinguish binary and decimal numbers, we add a capital ‘B’ to binary numbers. This way we can distinguish for example 11 (decimal eleven) from B11 (decimal 3), or 10 (decimal ten) from B10 (decimal 2).

Ok, that is enough preparation for now. We will collect more details as we work through the experiments. Let’s get started.

Exploration through Experimentation

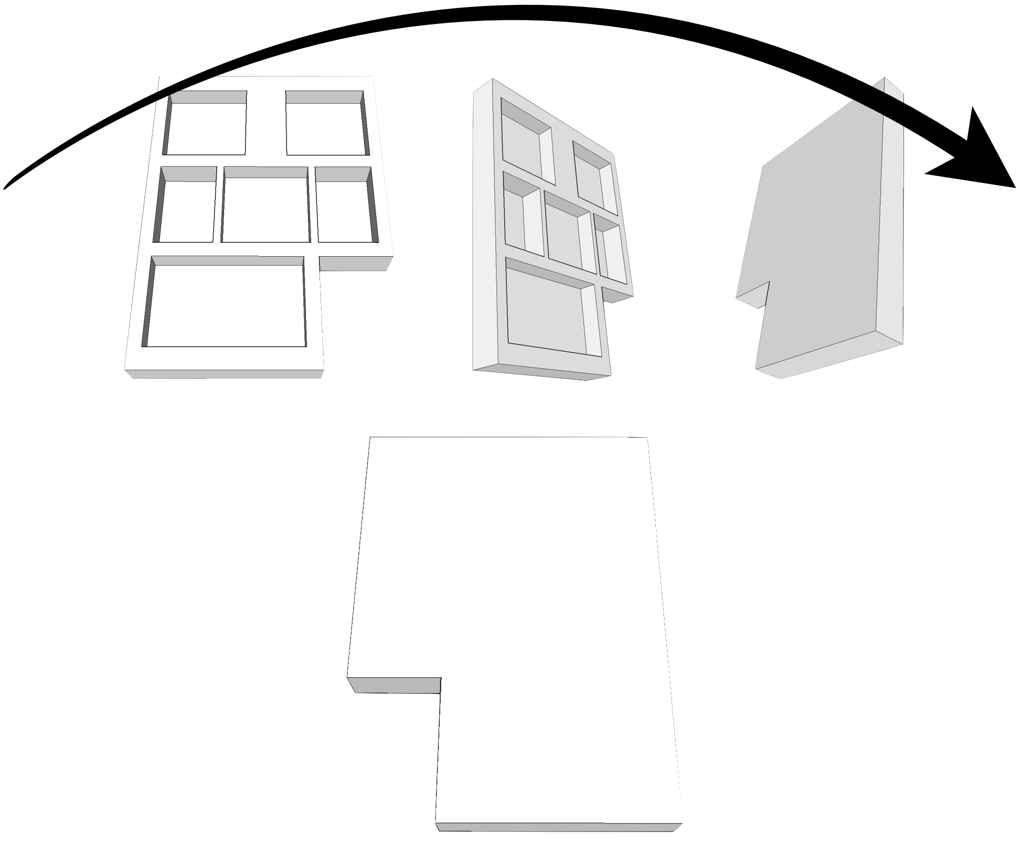
The B4 is an exploration kit. In this book, we explore through experimentation. Yes, we conduct experiments during which we will be plugging wires into the B4 modules, let them work together and experiment with data and hardware. On occasion, when the bell rings at the end of the lesson, we may not quite be finished with an investigation. So that we don't have to take all our good work apart (and start from scratch next experiment), the B4's packaging also serves as a storage tray. The foam insert contains several cut-outs to keep the B4's modules safe during transport. But now that the B4 has arrived, we no longer need them. Let's turn the packaging into a lab:

Step 1
Remove all the B4 modules and wires from the packaging and place them on your desk.

Step 2
Remove the foam insert from the box, flip it over - as shown below - and re-insert it with the flat side up. Reinsert it into the box.

Step 3
Place all modules neatly side-by-side on the foam insert and place all the wires in the cut-out at the top. This will keep the wires that we don't require for an experiment neatly in one place.

In the future, we can simply leave our experiments on the foam insert, close the lid, and place the box on a shelf - or anywhere else your teacher tells you.



Experiments

Overview

In this handbook, we have prepared several experiments that will help you to get to know the modules of the B4, how you use them and what functions they perform. Most experiments consist of one or more exercises. You will learn how to combine modules to that they perform functions together, which they could not perform individually.

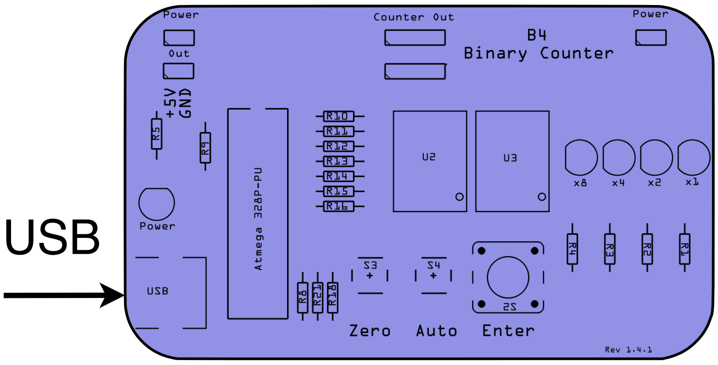
We recommend that you take the experiments in sequence. But if you are already a computer genius, feel free to jump around. We should mention that the B4 can do much more than what this handbook says. Feel free to explore and try out different things as you like.

Experiment	Title	Learning Objectives
1	We Learn to count in Binary	Function of the Binary Counter module. Binary number system. Binary counting.
2	Variables	What are variables in a computer? What is their function and how can they be used?
3	Converting Binary Numbers to Decimal	Internally, computers work with the binary number system, but humans prefer decimal numbers. In this experiment, students learn how to convert binary data to decimal.
4	Adding Two Numbers	Binary Addition. Adding as a fundamental computational principle. Limitations of computers: They can be wrong, even if they work perfectly well.
5	Adding Three Numbers	The adder module can only add two numbers. How can we add 3 numbers? 4? 5? ... (requires multiple kits)
6	Producing Graphics Output	Getting to know the Dot Matrix Display. Accessing letters, numbers and symbols.
7	Making Animations	Displaying images in rapid succession creates the illusion of a moving object.
8	Pixels	What are pixels? Free drawing on the Dot Matrix Display

Experiment 1: We learn to count in binary

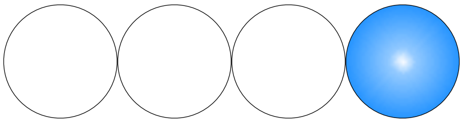
Modules Required: Binary Counter

Take the Binary Counter out of the box and connect it via a USB cable to your PC, laptop or battery.



Setup of Experiment 1

The Binary Counter only draws a little bit of electricity. With it, it will power itself and all the other B4 modules, as we will see later. The red display will show a ‘b4’ for B4. On the Binary Counter module, you will see a switch labelled *Enter*. Go ahead and press it. What happens? The display in the middle switches to 0. Press the button again and you will see a 1 in the middle and the right LED light comes on:

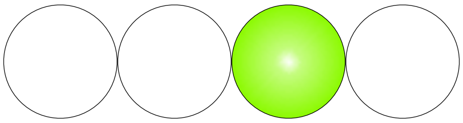


1 in decimal is 0001 in binary

Every time you press the Enter button, the number increases by 1.

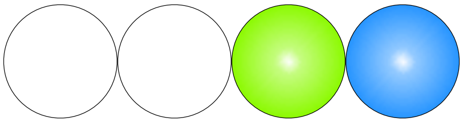
On the right hand side of the Binary Counter, you will see 4 LEDs. They show exactly the same number as the display, but in binary.

If your display shows a 2, then the LEDs will show a off-off-on-off, like this:



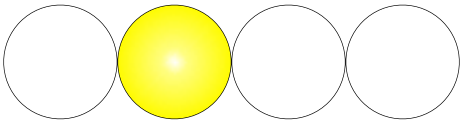
2 in decimal is 0010 in binary

If you press the button again, the display will show a 3 and the LEDs will show the following pattern:



3 in decimal is 0011 in binary

So, 3 is equal to 2+1. Which pattern will be displayed when you press the button again? The display shows a 4 and the LEDs will look like this: 0100



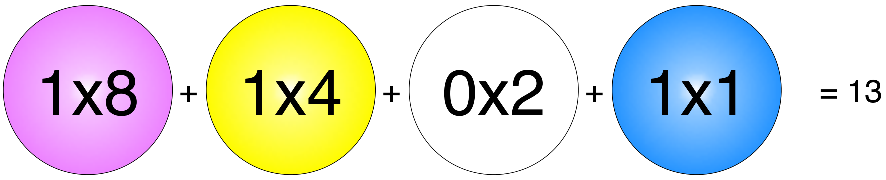
4 in decimal is 0100 in binary

If we keep pressing the button, we will see more light patterns in our LEDs and the corresponding decimal numbers on the display. We can enter these into a table.

binary	**decimal**
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

That's a lot of binary numbers. We could try to remember them by heart, but let's see if there is an easier way. Can you perhaps see a pattern in the table on the previous page?

Each of the LEDs stands for a particular number:



The blue LED stands for a 1. The green LED stands for a 2, the yellow LED represents a 4 and the pink LED stands for an 8. So, the number we see in the picture above is 8+4+1=13.
So, instead of remembering 16 different binary numbers, we only need to remember the decimal value that each LED represents, and we can then quickly calculate the number in our heads. What, you think there is an even easier way? Yes, you are right: These numbers double from right to left: 1, 2, 4, 8. So you only need to remember that the right LED represents a 1 and that the numbers double as we go from right to left. This means that we only have to remember **two** rules about binary numbers.

Question 1	
?	What is the decimal value of 1111?
	What is the decimal value of 0110?
	What is the decimal value of 1010?
	What is the binary value of decimal 15?
	What is the binary value of decimal 12?
	What is the binary value of decimal 9?
	How can you easily spot an odd binary number?

To this point we have learned that the Binary Counter counts from 0 to 15, or, in binary numbers, from B0000 to B1111. By now you have probably discovered that it will tick over to 0000 after 1111. Why is this? Remember that our counter is binary and it is 4 bit only. Adding 1 to B1111 results to a 5 bit number, which is B10000. And because our little counter can't store the 5th bit, it will simply think that B0000 is the new number. This is not a bug.

Experiment 2: Variables

Modules Required: Binary Counter, 1x Variable

Analyse

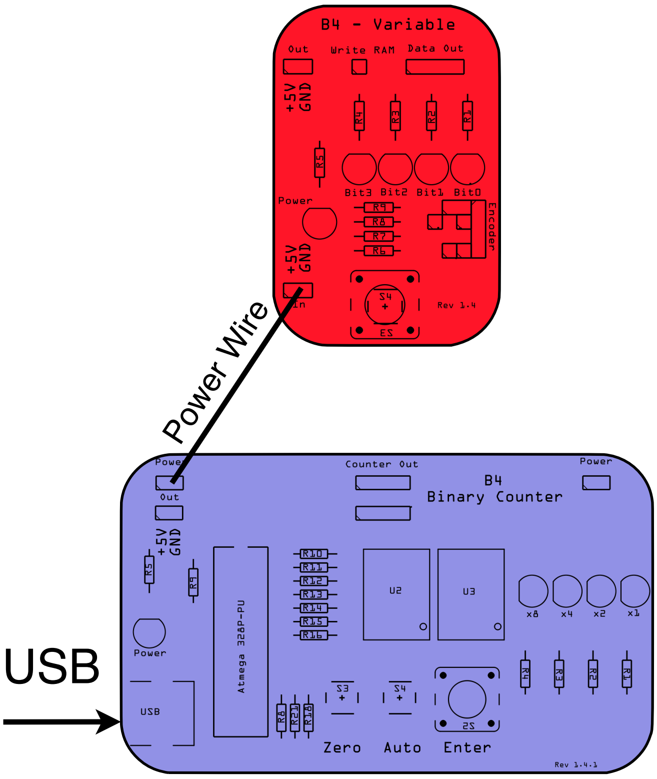
Variables are super-important in computing. They hold a piece of information, such as a number and make it available to the computer programs. In this experiment, we are going to get to know variables.

Build

The following diagram shows the setup of this experiment.

1) Place the Binary Counter and one Variable Module in front of you as shown below
2) Take a Power Wire
3) Connect the Power Wire to the Binary Counter
4) Connect the other end of the Power Wire to the Variable.
5) Connect the USB Cable to the Binary Counter
6) Connect the other end of the USB Cable to a Laptop.

The Binary Counter should now show ‘b4’ on its red display and the green light of the Variable should be on.



Setup of Experiment 2

Experiment
Turn the knob on the Variable left or right.

Observe
How do the LEDs on the Variable change?

Compare
The LEDs on the Variable Module will go on and off as you turn the knob. They will remain unchanged when you don't turn the knob.

Understand
The knob is able to make binary data. This binary data looks just like the binary data that the Binary Counter produced in Experiment 1.

Question 2	
?	Set the variable to any value, let's say 10 (B1010). Then, disconnect the USB power from the Binary Counter. Will the Variable remember its value when you reconnect the USB power?

Experiment 3: Converting binary numbers to decimal

Modules Required: Binary Counter, 1x Variable, Decimal Display

Analyse

Because we are more familiar with decimal numbers we want to convert binary numbers to decimal. Whilst we do this we find out a bit more about how the Variable works. For example, we learn that the contents of the Variable can be used by the Decimal Display.

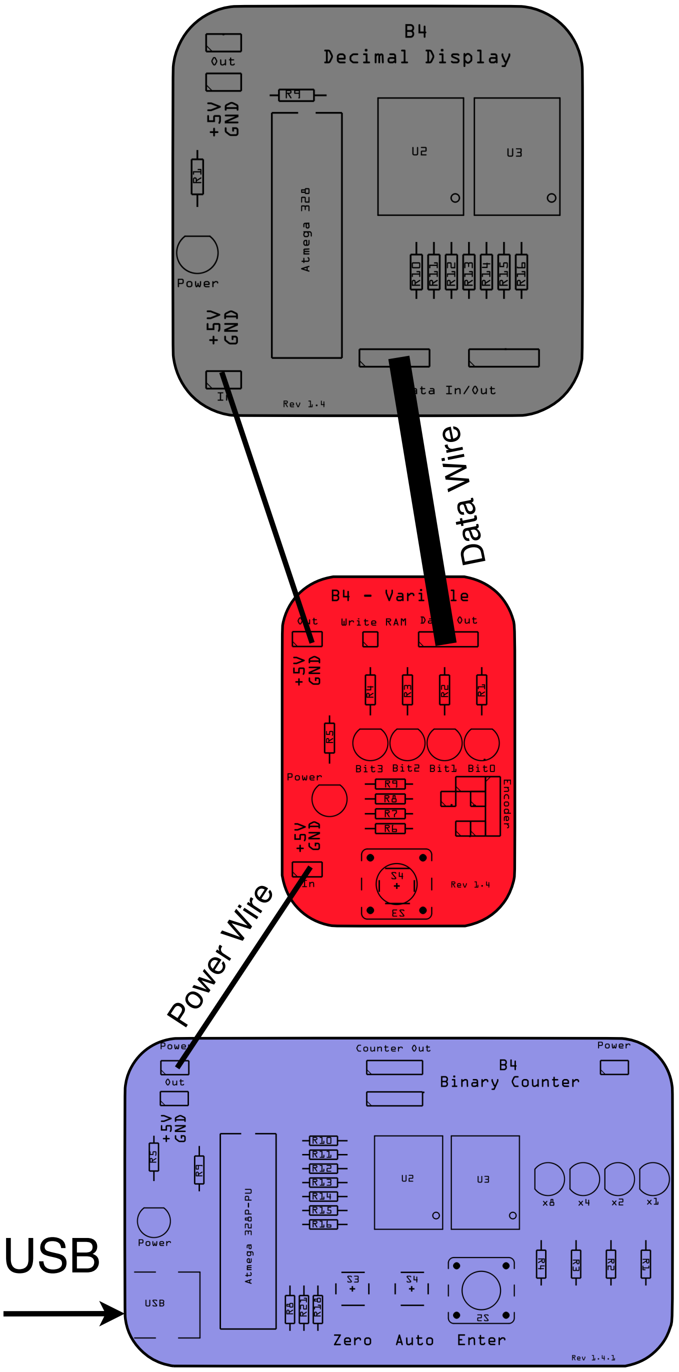
Build

Let's extend our build from the previous experiment:

1) Place the Decimal Display above the Variable Module as shown in the following figure on the next page.
2) Take a Power Wire
3) Connect the Power Wire to the Variable
4) Connect the other end of the Power Wire to the Decimal Display.
5) Take a Data Wire
6) Connect the Data Wire to the Variable
7) Connect the other end of the Data Wire to one of the 4 pin ports of the Decimal Display. It doesn't matter which one you choose.

The green light of the Decimal Display should be on and it should show some number on its red 7 segment LEDs.

The following diagram shows the setup of this experiment.



Setup of Experiment 3

Experiment
1) Turn the knob on the Variable to the right in clock direction. 2) Turn the knob on the Variable to the left in anti-clock direction.

Observe
How do the numbers on the Decimal Display change?

Compare

The numbers will grow bigger when you turn the knob on the Variable to the right and will get smaller when you turn the knob to the left.

Understand

We learn four things from this experiment:
1) Turning the knob on the variable produces binary numbers
2) These numbers grow as we turn the knob on the variable to the right
3) The numbers shrink as we turn the knob to the left (anti-clock)
4) The Decimal Display translates binary numbers to decimal numbers

Question 3	
?	What is the biggest number that the variable can produce? What is its binary value?
	What is the smallest number that the variable can produce? What is its binary value?
	Set the variable to 15 (that’s B1111) and then turn the knob to the right for one notch. What is the output of the variable? Can you explain it?

Experiment 4: Adding two numbers

Modules Required: Binary Counter, Adder, 2x Variable

Analyse

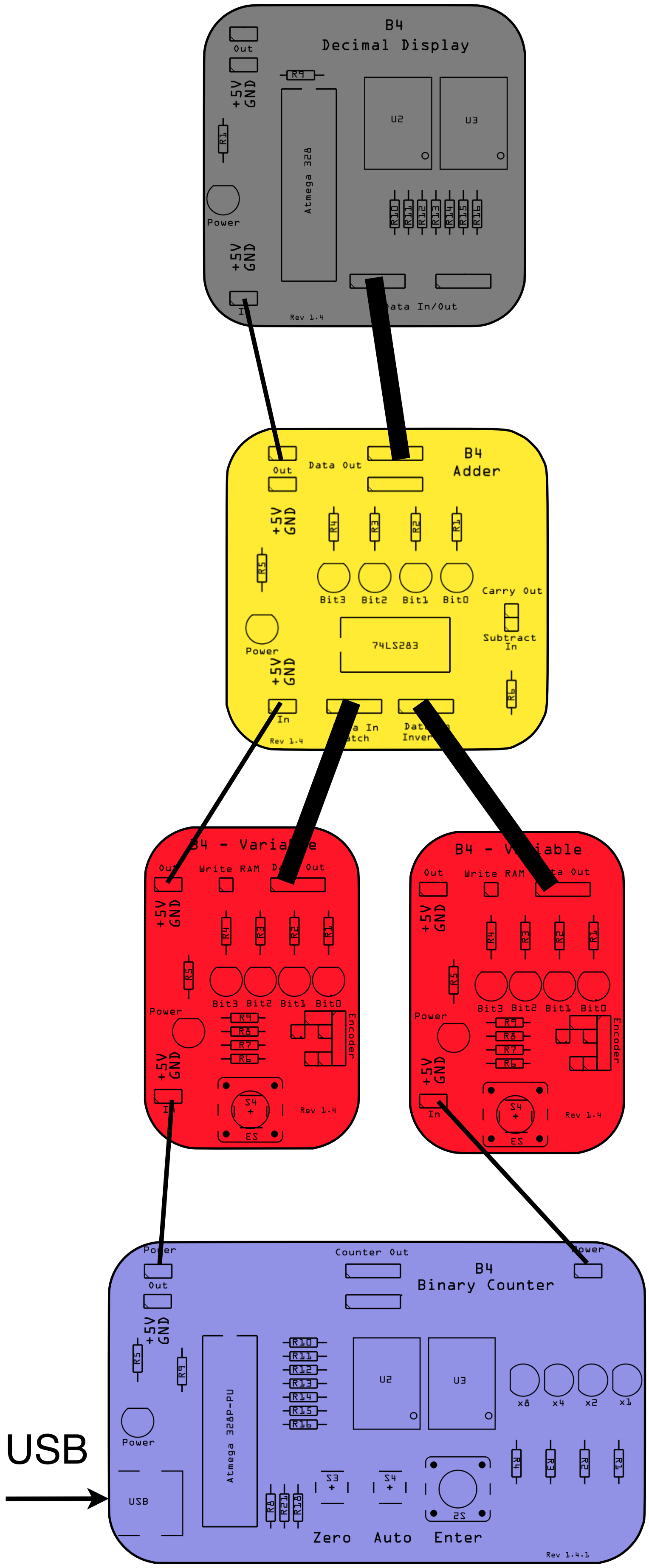
Adding numbers is the most fundamental thing a computer does. When you know how to add, you can also do multiplication, because 3x5 is just a series of additions, namely 5+5+5. So, by adding repeatedly, a computer can do multiplication. Addition is not only fundamental to computers. At school, in mathematics, students learn addition well before subtraction, multiplication and division.

Build

1) Place the modules in front of you as shown in the following figure.
2) Connect all 4 Power Wires.
3) Connect all 3 Data Wires
4) Connect the USB cable from the Binary Counter to your Laptop Computer.

With the Variables, we will set numbers that we want the Adder to add up. In this experiment, we also require the Binary Counter, but only to provide electricity from your Computer's USB port to the Variables and to the Adder module.

The following diagram shows the setup of this experiment.



Setup of Experiment 4

Experiment

1) Turn the knobs on both Variables until their LEDs are all off.
2) Turn the knob on one of the Variables so that just the right most LED turns on.

Observe

What is the output of the Adder?

Compare

The right LED on the Adder will light up.

Understand

That’s because 0+1 is 1.

Turn the knob of the other Variable to show the 0001 LED pattern.

The Adder will then show 0010. That’s because 1+1=2, which is 0010 in binary. Binary addition works just like the addition you already know with one difference, any number higher than 1 leads to a carry over. In the decimal number system that you already know, any number higher than 9 leads to a carry over. So, in a sense, binary addition is simpler than decimal addition.

```
  0001     0101
+0001    +0110
--------  --------
  0010     1011
```

Question 4	Compute with your B4 and also on paper:
?	What is B0101 + B1010?
	What is B0010+B0010?
	What is B0111+B0001?
	What is B1111 + B0001? Why are all the Adder’s LEDs off?

Experiment 5: Adding three numbers

Modules Required: Binary Counter, 2xAdder, 3xVariable

In the previous experiment, we added two numbers. But what if we want to add three numbers, such as 1+2+3? For this, we need another Adder Module and Variable, so best team up with another student who has another B4.

Analyse

Our Adder has only two input ports and can always only add two numbers, not three, four, or five. However, because we want to add three numbers, we apply a little mathematical trick, which is called the associative law.
The law states that we more than two numbers in any order. So 1+2+3 is the same as first adding 1+2 and then adding 3 to the sum. The total is always 6.

Let's try this

5+2+1 = 7+1 = 8
6+2+5 = 8+5 =13

In fact, it works for any number which we want to add.

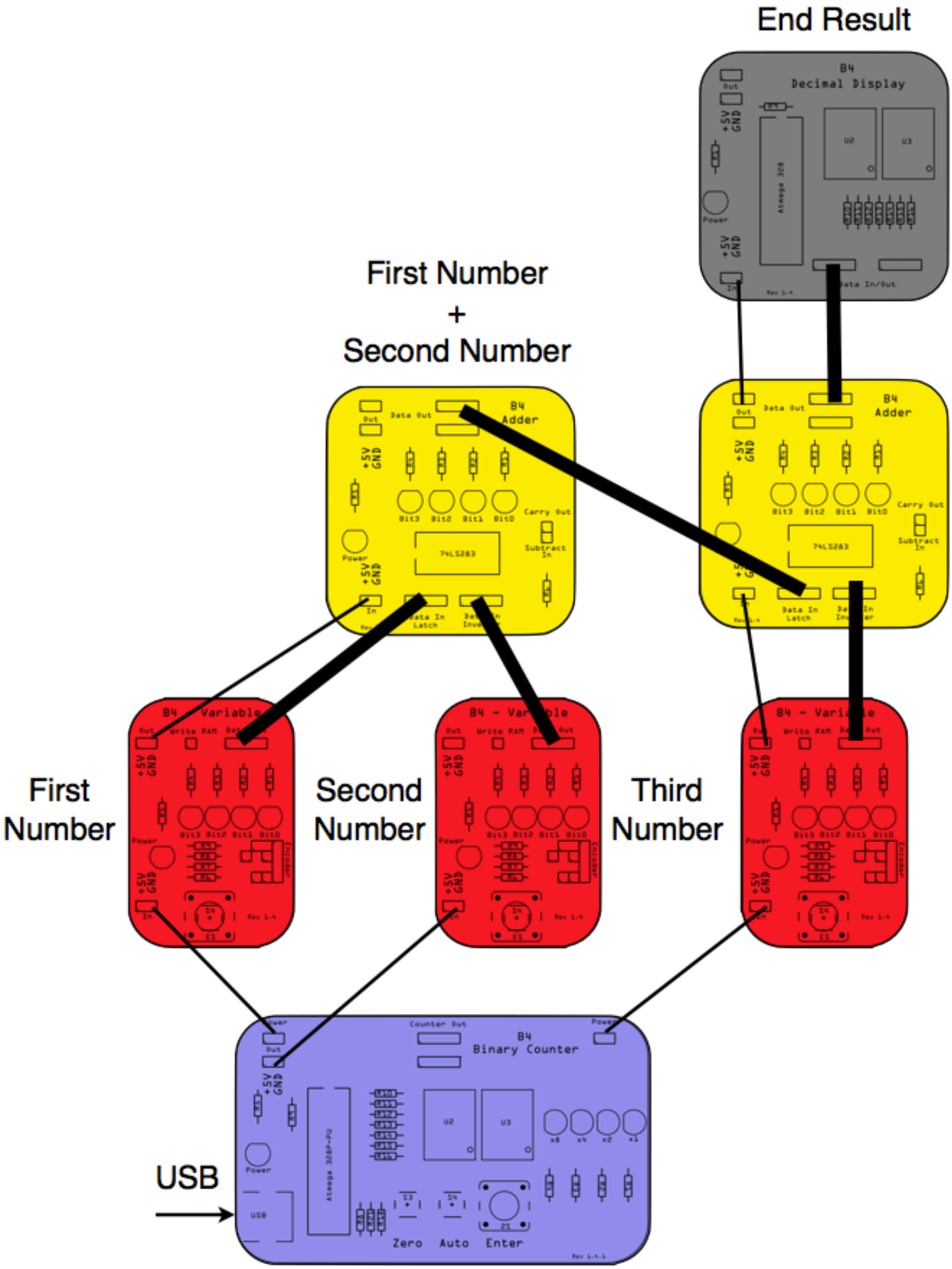
To make this happen with our B4, we want to use two Adders for the two additions and three Variables to hold the three numbers.

Build

1) Place the modules in front of you as shown in the following figure.
2) Connect all 5 Power Wires.
3) Connect all 4 Data Wires
4) Connect the USB cable from the Binary Counter to your Laptop Computer.

The following diagram shows the setup of this experiment.

We have labelled the function of the modules. With the three variables, we will set the three numbers that we want to add. The left Adder will add the first and second numbers. The *intermediate result* will then flow to the right Adder, which will add it to the third number coming from the right variables. The Decimal Display will then show the end result.



Setup of Experiment 5

Experiment 5.1

Set all Variables to B0000 (Zero)

Observe 5.1

What is the sum shown on the Decimal Display?

Compare 5.1

It is 0 (zero).

Understand 5.1

0+0+0=0

Experiment 5.2

Using the Variables:
1) Set the first number to B0001 (one)
2) Set the first number to B0010 (two)
3) Set the first number to B0011 (three)

Observe 5.2

What is the sum shown on the Decimal Display?

Compare 5.2

It is 6 (six).

Understand 5.2

1+2=3, which is the intermediate result
3+3=6

Experiment 5.3

Using the Variables:
1) Set the first number to B0101 (five)
2) Set the first number to B0010 (two)
3) Set the first number to B0001 (one)

Observe 5.3

What is the sum shown on the Decimal Display?

Compare 5.3

It is 8 (eight).

Understand 5.3

5+2=7, which is the intermediate result
7+1=8

Question 5	Using your B4, compute:
?	10+3+2
	1+1+1
	2+3+8
	3+2+1
	0+0+1

Experiment 6: Producing graphics output

Modules Required: Binary Counter, Variable, Dot Matrix Display

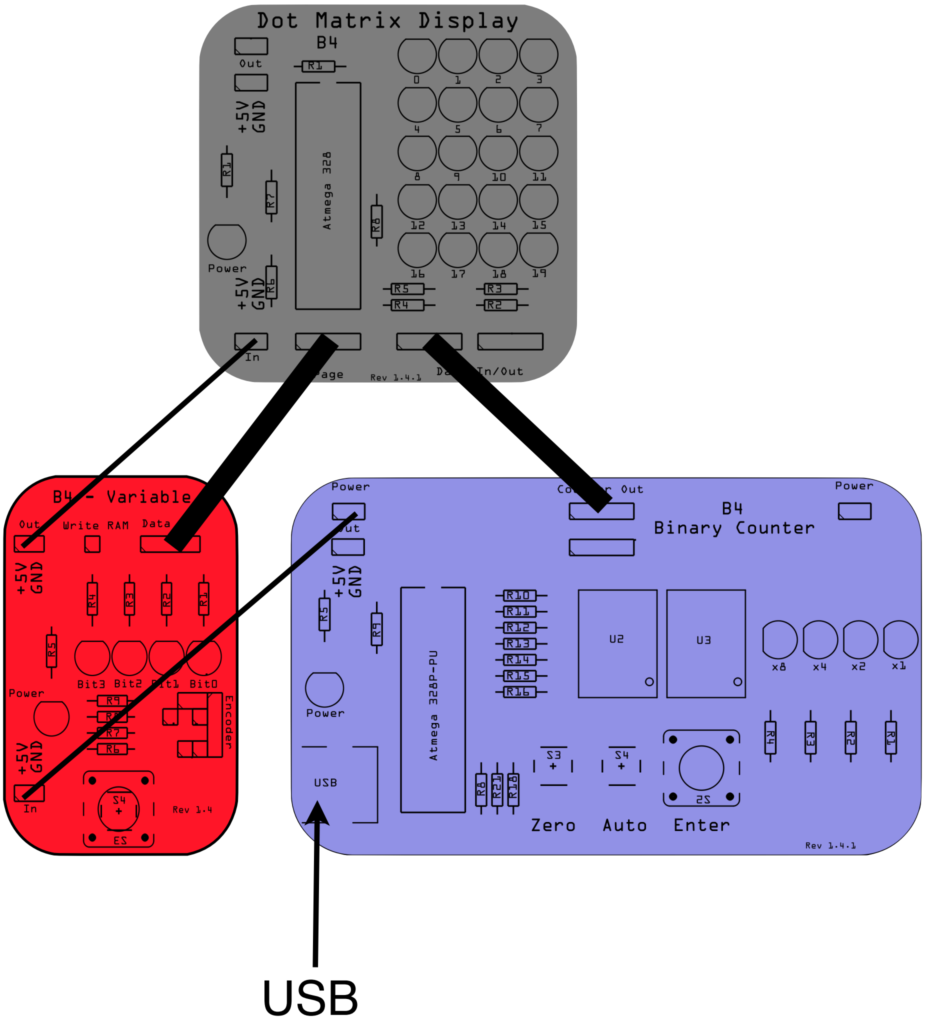
Analyse

Our aim is to get to know the Dot Matrix Display. Since it has two inputs, called mode and data, we provide binary data to these ports and observe what happens.

Build

1) Setup and Power
    - A. Place the three modules on the desk in front of you as shown in the following diagram.
    - B. Connect the Variable to the Binary Counter with power wires as shown.
    - C. Use a power wire and connect the Variable with the Dot Matrix Display.
2) Data
    - A. Connect the 4 pin output of the Variable to the Page connector of the Dot Matrix Display.
    - B. Connect the 4 pin output of the Binary Counter to the Data In/Out port of the Dot Matrix Display. There are two Data In/Out ports on the Dot Matrix Display. You can use either. The second one is useful when we want to mirror the output to a second Display. This is not important at the moment. Just keep it in mind.

The following diagram shows the setup of this experiment.



Setup of Experiment 6

Our first letter: A

Experiment 6.1
1) Connect the Binary Counter to a USB power source. 2) Set the variable to B0000.

Observe 6.1
What is the output of the Dot Matrix Display?

Compare 6.1
The Display should now show the letter ‘A’.

Our second letter: B

Experiment 6.2
Click the Enter button on the Binary Counter so that it shows B0001.

Observe 6.2
What is the output of the Dot Matrix Display?

Compare 6.2
The Display should now show the letter ‘B’.

Our third letter: C

Experiment 6.3
Click the Enter button on the Binary Counter again so that it shows B0010.

Observe 6.3
What is the output of the Dot Matrix Display?

Compare 6.3
The Display should now show the letter ‘C’.

More letters: D-P

Experiment 6.4
Continue incrementing the binary Counter until all LEDs are on, which is B1111.

Observe 6.4
What is the output of the Dot Matrix Display?

Compare 6.4
The Display should now show the letter ‘P’.

Evaluate

“Hang on”, you might say. “The alphabet doesn’t end with ‘P’. There should be more letters”. Rightly so. You remember that the B4 is a 4 bit computer and thus limited to the numbers 0 to 15? Our alphabet, however, has 26 different characters. How can we distinguish 26 different characters with 4 bit? We could add a 5th bit to the B4, but this would mean changing every single module. Another way is to use two 4 bit numbers. And this is where the left Variable comes into play. So we have created pages of 16 characters each. The left Variable will choose the page and the Binary Counter will select the character, number or symbol from within that page. So, the first character on page 0 is an ‘A’. We have already seen this.

Let’s make a Q

Experiment 6.5

Set the binary Counter back to B0000 and the left Variable to B0001 (page 1)

Observe 6.5

What is the output of the Dot Matrix Display?

Compare 6.5

The Display should now show the letter ‘Q’.

R comes after Q

Experiment 6.6

Set the Binary Counter to B0001 and leave the left Variable to B0001 (page 1)

Observe 6.6

What is the output of the Dot Matrix Display?

Compare 6.6

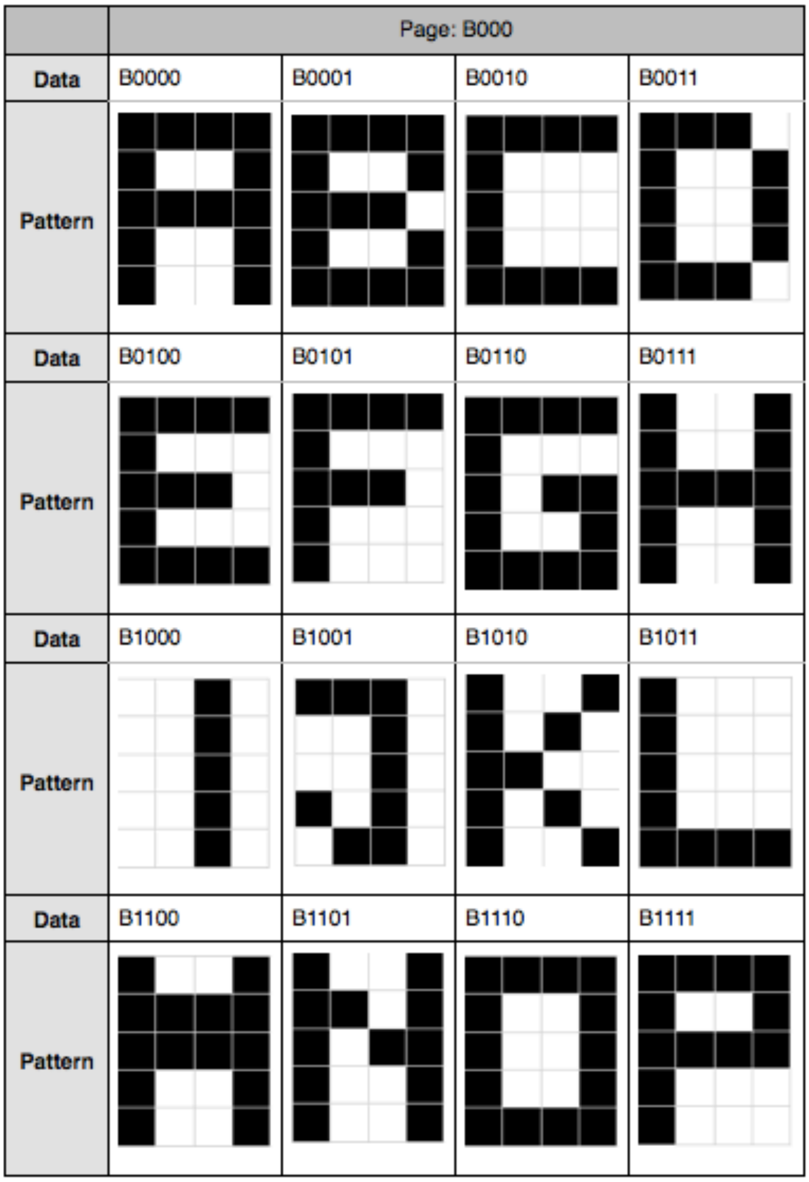
The Display should now show the letter ‘R’.

Evaluate

The Dot Matrix Display has 10 pages of letters, numbers and symbols. You will find each of them in Appendix A. The following table provides a brief overview of what you get from each page:

Page Number	Content
B0000	Letters A ... P
B0001	Letters Q ... Z
B0010	Numbers 0 ...15
B0011	Symbols, including emojis
B0100	Flying Arrow
B0101	Dancing Dots
B0110	Pixel Mode
B0111	Ones digit of input number
B1000	Tens digit of input number
B1111	Display diagnostics

Summary of the Dot Matrix Display Output



	Page: B000			
Data	B0000	B0001	B0010	B0011
Pattern	■■■■ ■□□■ ■■■■ ■□□■ ■□□■	■■■■ ■□□■ ■■■□ ■□□■ ■■■■	■■■■ ■□□□ ■□□□ ■□□□ ■■■■	■■■□ ■□□■ ■□□■ ■□□■ ■■■□
Data	B0100	B0101	B0110	B0111
Pattern	■■■■ ■□□□ ■■■□ ■□□□ ■■■■	■■■■ ■□□□ ■■■□ ■□□□ ■□□□	■■■■ ■□□□ ■□■■ ■□□■ ■■■■	■□□■ ■□□■ ■■■■ ■□□■ ■□□■
Data	B1000	B1001	B1010	B1011
Pattern	□□■□ □□■□ □□■□ □□■□ □□■□	■■■□ □□■□ □□■□ ■□■□ □■■□	■□□■ ■□■□ ■■□□ ■□■□ ■□□■	■□□□ ■□□□ ■□□□ ■□□□ ■■■■
Data	B1100	B1101	B1110	B1111
Pattern	■□□■ ■■■■ ■■■■ ■□□■ ■□□■	■□□■ ■■□■ ■□■■ ■□□■ ■□□■	■■■■ ■□□■ ■□□■ ■□□■ ■■■■	■■■■ ■□□■ ■■■■ ■□□□ ■□□□

Characters A ... P on *Page 0 (see Appendix A)*

Let’s make more letters, numbers and symbols

Experiment 6.7
Go and experiment with the Variables and try to generate each pattern from Appendix A on the Dot Matrix Display. Have fun !

Evaluate Deeper

How does it work?
When we send the page and data values to the Dot Matrix Display, a little microprocessor will decide which pattern to draw. It will lookup the pattern in its program and send electric impulses to the LEDs. Those that receive electricity will then light up.

So we are not actually sending a character to the Dot Matrix Display, but a reference to the character, which is already stored in the memory of the Dot Matrix Display. This is called a **Data Pointer**. The processor on the Dot Matrix Display is called a **Graphics Processor**, because it deals exclusively with graphics content.

Question 6	Try to find the following characters in the Dot Matrix Display. The pages in Appendix A will be helpful.
?	A happy smiley
	A question mark
	A dot
	A closing bracket
	The number 8

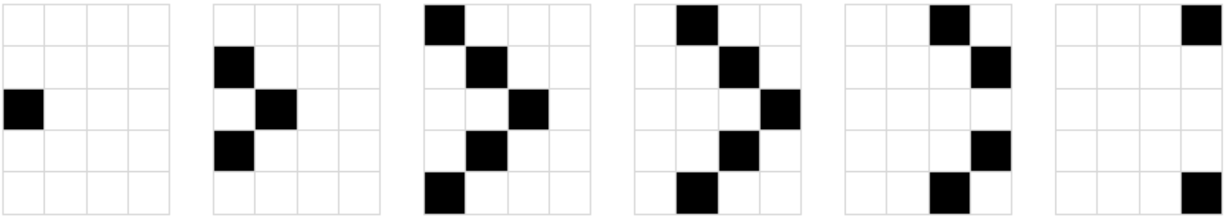
Experiment 7: Making animations

We use the same setup from the previous experiment 6.

Analyse

Have you ever seen a movie? Of course you have. Did you know that a movie is just a sequence of images that are shown very quickly?

Can you produce an animation of an arrow that flies through the display from left to right? We find the necessary characters on page **B0100** of the Dot Matrix Display. Here they are:



Build 7.1

1) Re-use the setup from the previous experiment and set the Variable to **B0100**. With this, we get access to the characters we need for this experiment.
2) Set the Binary Counter to 0 by pressing the Zero button.

Experiment 7.1

Press the Enter button on the Binary Counter repeatedly.

Observe 6.1

What is the output of the Dot Matrix Display?

Compare 6.1

The Display should now show the characters that you see at the top of this page, namely an arrow that flies across the screen.

Experiment 7.2

Can we make the characters fly faster? Yes we can !
On the Binary Counter, press and hold the **Auto** button.

Observe 7.2

What is the output of the Dot Matrix Display?

Compare 7.2

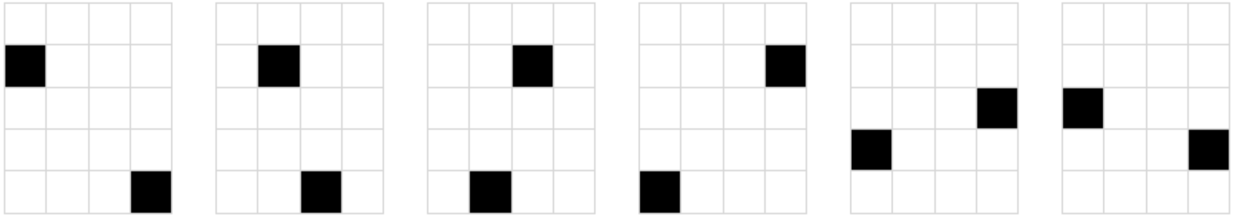
The characters on the Dot Matrix Display change rapidly.

Evaluate

Our eyes and brain are just not fast enough to see them individually and thus create the illusion of a ‘motion picture’. We have made our first computer animation.

Analyse

Let’s now make two dots circle each other. Take a look into Appendix A. There, you find the necessary characters on page **B0101** of the Dot Matrix Display. Here they are:



Build 7.3

1) Re-use the setup from the previous experiment and set the Variable to **B0101**. With this, we get access to the characters we need for this experiment
2) Set the Binary Counter to 0 by pressing the Zero button.

Experiment 7.2

On the Binary Counter, press and hold the Auto button.

Observe 7.2

What is the output of the Dot Matrix Display?

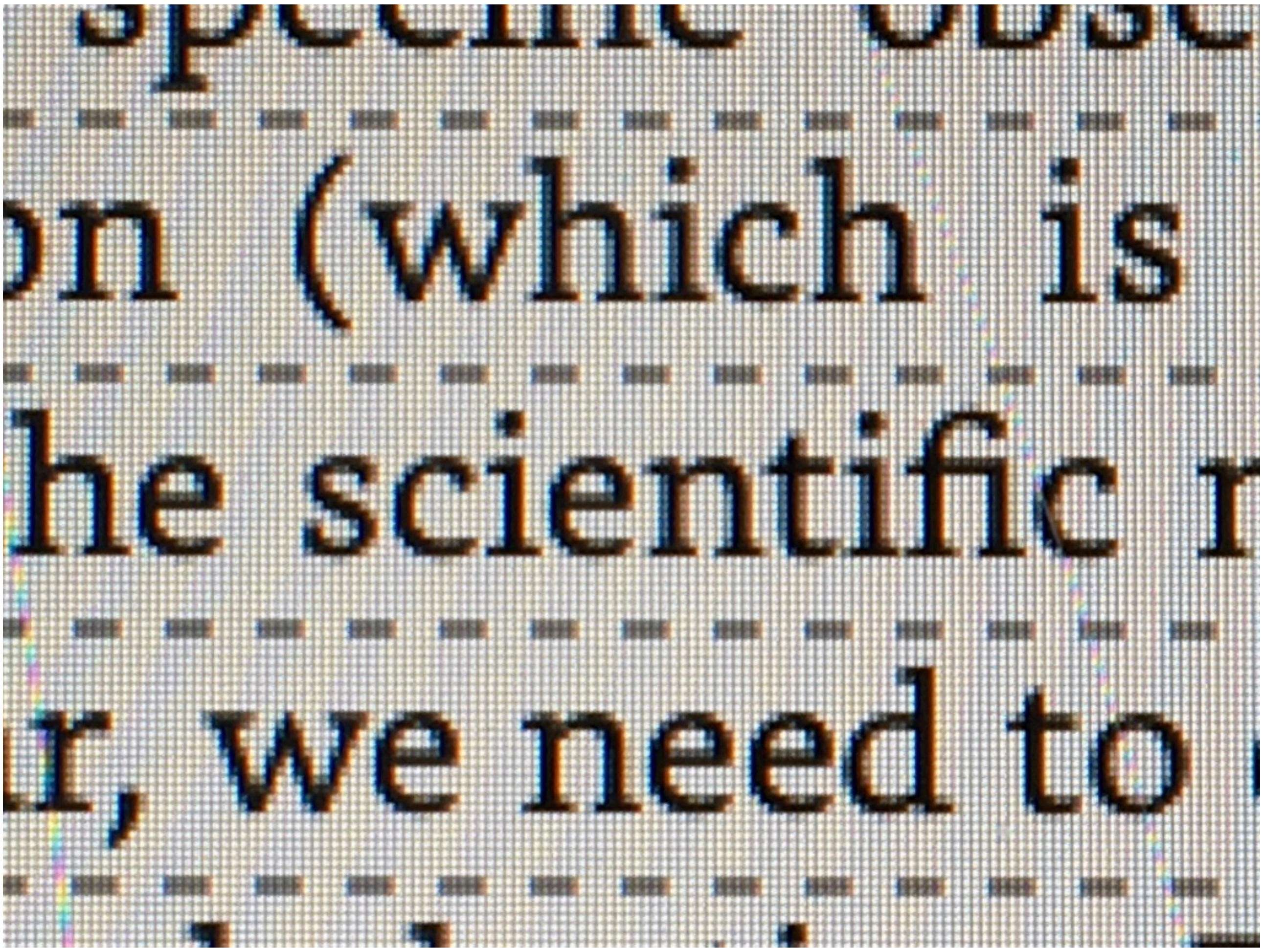
Compare 7.2
The characters on the Dot Matrix Display change rapidly. You should see an animated square.

Question 7	
?	Experiment a bit further by changing the value of the Variable to B0000, B0001, B0002, etc. What other interesting animations can you produce? Have fun!

Experiment 8: Pixels

Analyse

In the previous experiment, we used the various characters that have been pre-programmed into the Dot Matrix Module. However, the page **B0110** allows the direct access of individual LEDs of the Dot Matrix Display. Each of the LEDs in a display is called a pixel. According to our friends at Wikipedia, “The word pixel is based on a contraction of pix (from word "pictures", where it is shortened to "pics", and "cs" in "pics" sounds like "x") and el (for "element")[...]”[2]. A pixel is the smallest addressable point in a display device. All displays consist of pixels. The are so tiny that we can’t tell them apart with our naked eyes. However, when looked at under a magnifying glass, they do become visible. In the image below, we have taken a photo of an iPad 2 screen with an iPhone 6 camera. We can clearly see the grid of pixels on the iPad’s screen.



Pixels of an iPad 2 Display

Using the setup from the previous experiment, have a go and try to access the pixels of the Dot Matrix Display. We won’t show the experiment for this because we think you can do this yourself. Have a go :-)

Further Reading

Below we have listed some really good resources that we used during the design of the B4.

Charles Petzold, CODE The Hidden Language of Computer Hardware and Software, 1999
http://www.charlespetzold.com/code/
ASCII Table and Description http://www.asciitable.com
ASCII https://en.wikipedia.org/wiki/ASCII
Pixel: https://en.wikipedia.org/wiki/Pixel
Pixel Art https://en.wikipedia.org/wiki/Pixel_art

Troubleshooting

Every good experiment has the potential for failure. This is usually the moment when we learn something new. Below is a list of the typical errors and their solutions.

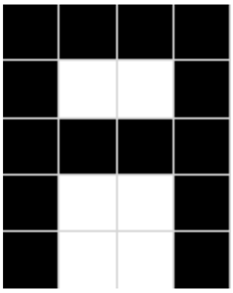
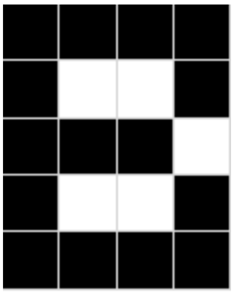
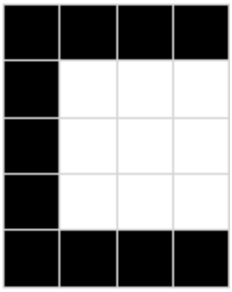
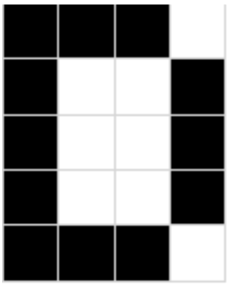
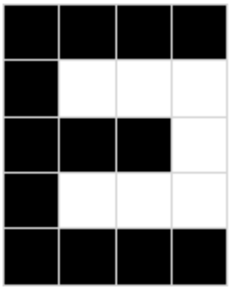
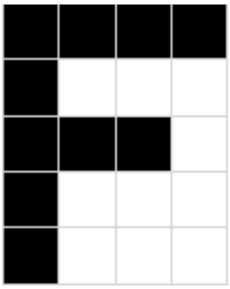
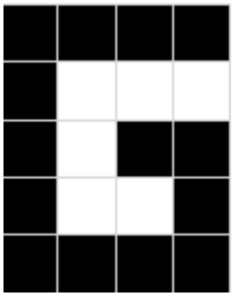
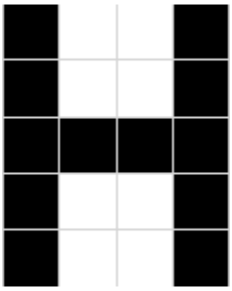
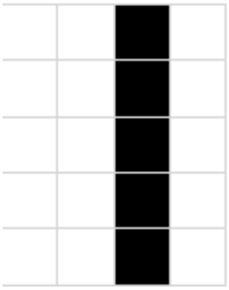
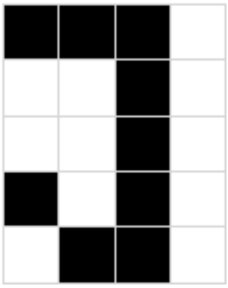
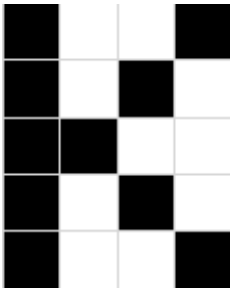
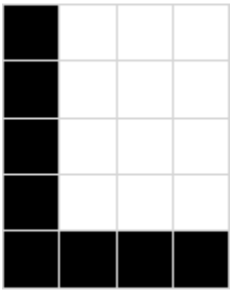
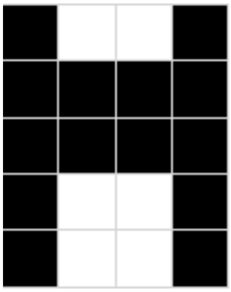
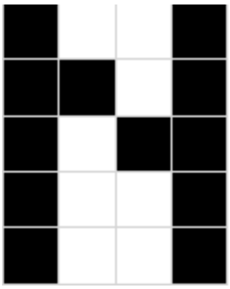
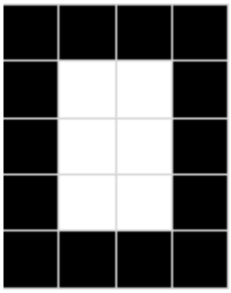
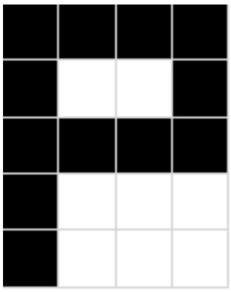
Symptom	Solution
Green light of a module is off	Check if power cable is connected
	Check if the wires at the power cable plugs are fully inserted. Change cable.
Unexpected behavior. Odd output of the modules. Looks erratic.	Check if all wires are properly connected. Tick them off one by one on the schematic of the corresponding experiment.
	Check if the wires at the data cable sockets are fully inserted. Change cable.
	Have you inserted the correct module? Check!
All lights are off	Connect USB cable to a computer, USB power outlet or USB battery.
	There may be a short circuit, usually cause by a power cable. Disconnect all power cables from the Binary Counter and check if the Binary Counter's green LED comes on. If yes, carefully connect one module after the other.

Still got problems? Email us at: enquiries@digital-technologies.institute.

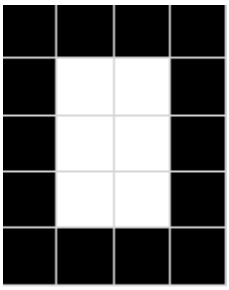
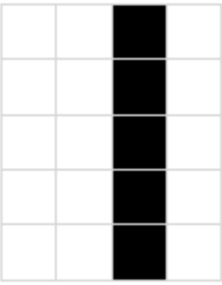
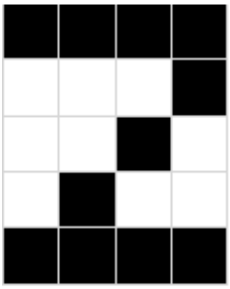
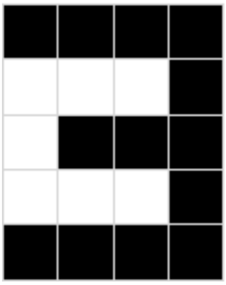
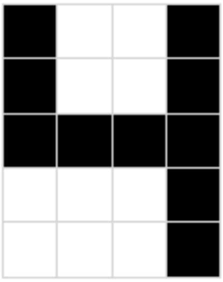
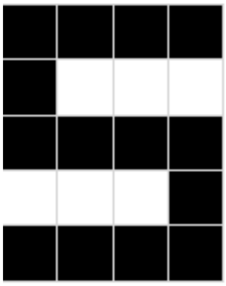
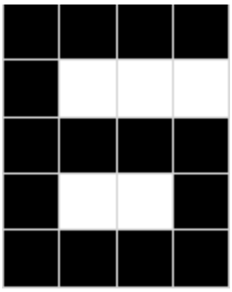
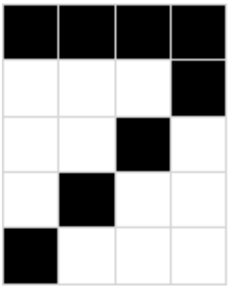
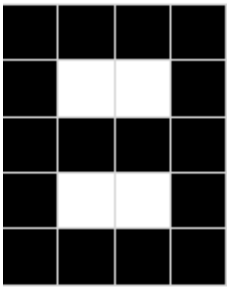
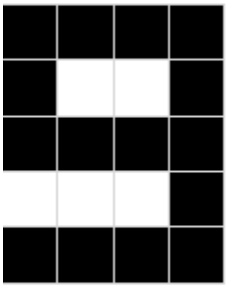
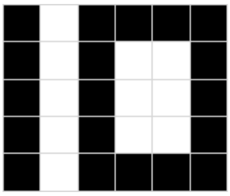
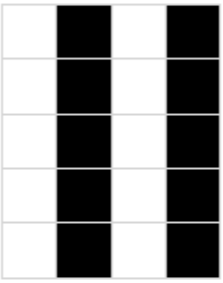
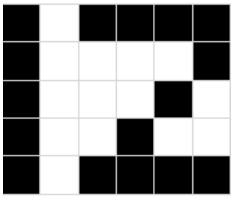
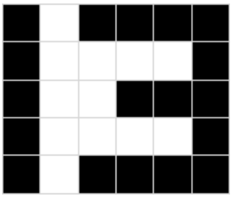
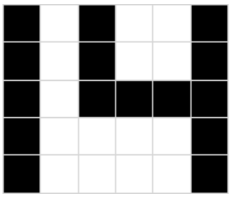
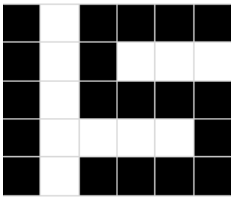
Appendix A: Dot Matrix Display Character Tables

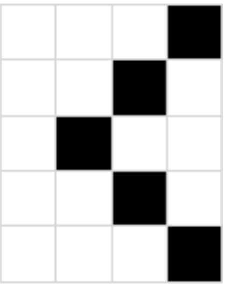
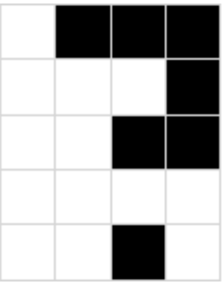
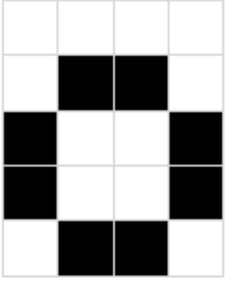
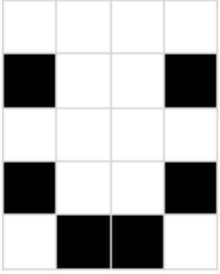
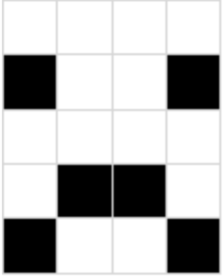
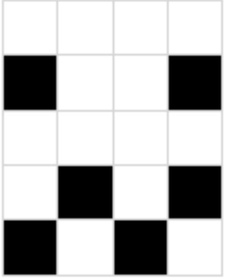
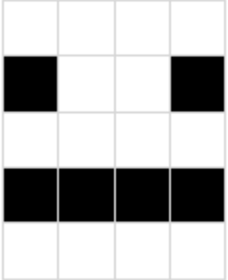
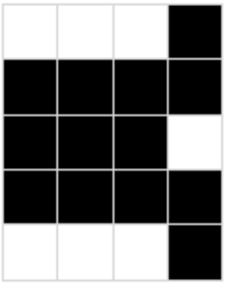
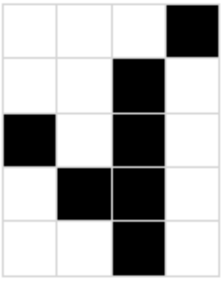
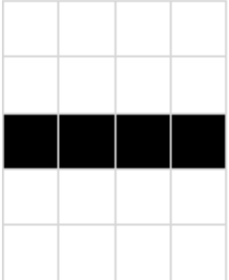
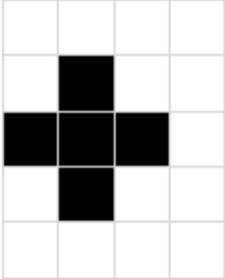
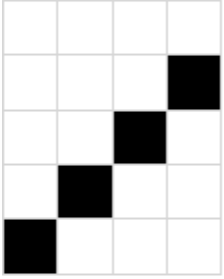
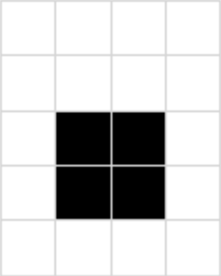
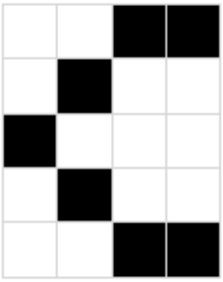
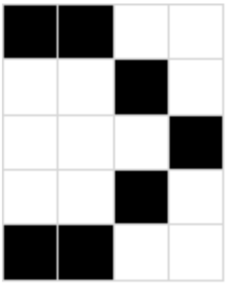
The Dot Matrix Display has 10 pages (B0000 to B0110) of letters, numbers and symbols. The following table provides a brief overview of what you get from each page. You will find each of them on the following pages.

Page Number	Content
B0000	Letters A ... P
B0001	Letters Q ... Z
B0010	Numbers 0 ...15
B0011	Symbols, including smileys
B0100	Flying Arrow
B0101	Dancing Dots
B0110	Pixel Mode
B0111	Ones digit of input number
B1000	Tens digit of input number
B1111	Display diagnostics

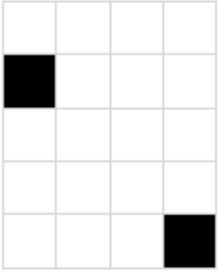
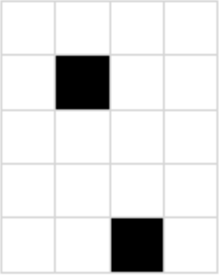
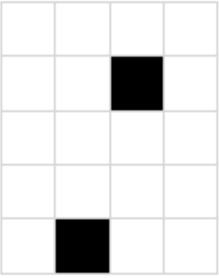
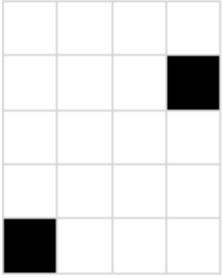
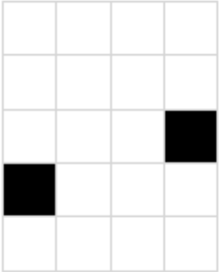
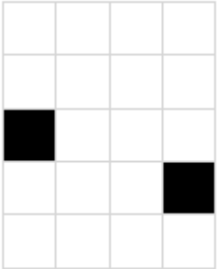
	Page B000: Letters A ... P			
Data	B0000	B0001	B0010	B0011
Pattern				
Data	B0100	B0101	B0110	B0111
Pattern				
Data	B1000	B1001	B1010	B1011
Pattern				
Data	B1100	B1101	B1110	B1111
Pattern				

	Page B0001 Letters Q ... Z			
Data	B0000	B0001	B0010	B0011
Pattern	■■■■ ■□□■ ■□□■ ■□■■ ■■■■	■■■■ ■□□■ ■■■■ ■□■□ ■□□■	■■■■ ■□□□ ■■■■ □□□■ ■■■■	■■■■ □□■□ □□■□ □□■□ □□■□
Data	B0100	B0101	B0110	B0111
Pattern	■□□■ ■□□■ ■□□■ ■□□■ ■■■■	■□□■ ■□□■ ■□□■ ■□□■ □■■□	■□□■ ■□□■ ■■■■ ■■■■ ■□□■	□□□□ ■□□■ □■■□ □■■□ ■□□■
Data	B1000	B1001	B1010	B1011
Pattern	■□□■ ■□□■ ■■■■ □□□■ ■■■■	■■■■ □□□■ □□■□ □■□□ ■■■■	□□□□ □□□□ □□□□ □□□□ □□□□	□□□□ □□□□ □□□□ □□□□ □□□□
Data	B1100	B1101	B1110	B1111
Pattern	□□□□ □□□□ □□□□ □□□□ □□□□	□□□□ □□□□ □□□□ □□□□ □□□□	□□□□ □□□□ □□□□ □□□□ □□□□	□□□□ □□□□ □□□□ □□□□ □□□□

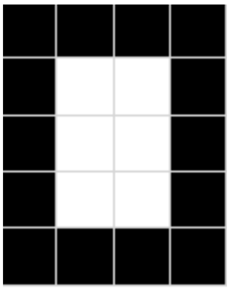
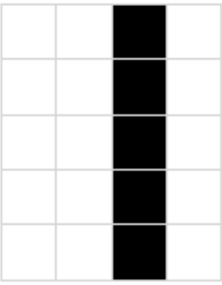
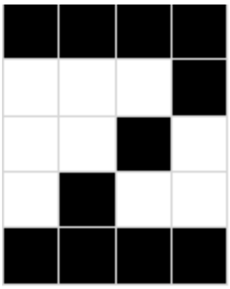
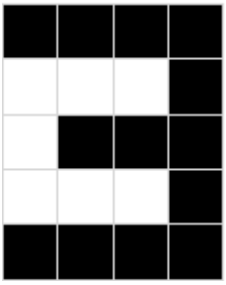
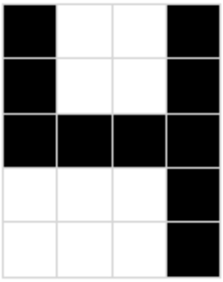
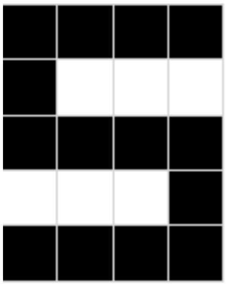
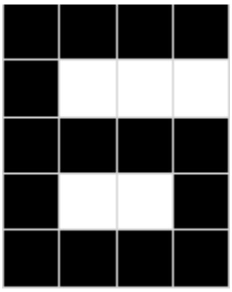
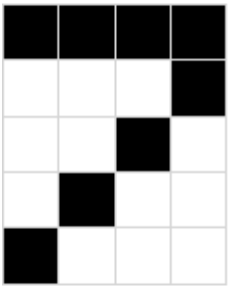
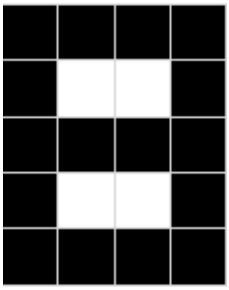
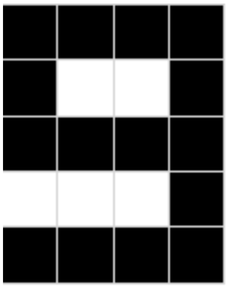
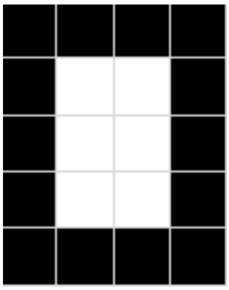
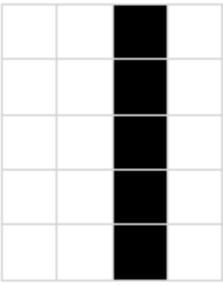
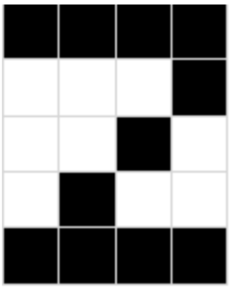
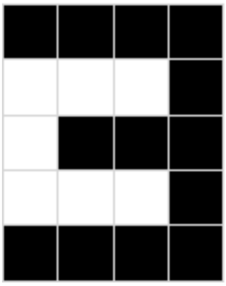
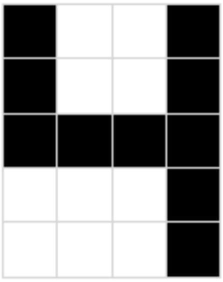
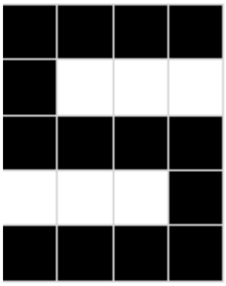
	Page B0010: Numbers 0 ...15			
Data	B0000	B0001	B0010	B0011
Pattern				
Data	B0100	B0101	B0110	B0111
Pattern				
Data	B1000	B1001	B1010	B1011
Pattern			 scrolling	
Data	B1100	B1101	B1110	B1111
Pattern	 scrolling	 scrolling	 scrolling	 scrolling

	Page B0011: Symbols			
Data	B0000	B0001	B0010	B0011
Pattern				
Data	B0100	B0101	B0110	B0111
Pattern				
Data	B1000	B1001	B1010	B1011
Pattern				
Data	B1100	B1101	B1110	B1111
Pattern				

	Page B0100: Flying Arrow			
Data	B0000	B0001	B0010	B0011
Pattern	□□□□ □□□□ ■□□□ □□□□ □□□□	□□□□ ■□□□ □■□□ ■□□□ □□□□	■□□□ □■□□ □□■□ □■□□ ■□□□	□■□□ □□■□ □□□■ □□■□ □■□□
Data	B0100	B0101	B0110	B0111
Pattern	□□■□ □□□■ □□□□ □□□■ □□■□	□□□■ □□□□ □□□□ □□□□ □□□■	□□□□ □□□□ □□□□ □□□□ □□□□	□□□□ □□□□ □□□□ □□□□ □□□□
Data	B1000	B1001	B1010	B1011
Pattern	□□□□ □□□□ □□□□ □□□□ □□□□	□□□□ □□□□ □□□□ □□□□ □□□□	□□□□ □□□□ □□□□ □□□□ □□□□	□□□□ □□□□ □□□□ □□□□ □□□□
Data	B1100	B1101	B1110	B1111
Pattern	□□□□ □□□□ □□□□ □□□□ □□□□	□□□□ □□□□ □□□□ □□□□ □□□□	□□□□ □□□□ □□□□ □□□□ □□□□	□□□□ □□□□ □□□□ □□□□ □□□□

	Page B0101: Dancing Dots			
Data	B0000	B0001	B0010	B0011
Pattern				
Data	B0100	B0101	B0110	B0111
Pattern				
Data	B1000	B1001	B1010	B1011
Pattern				
Data	B1100	B1101	B1110	B1111
Pattern				

	Page B0110: Pixel Mode			
Data	B0000	B0001	B0010	B0011
Pattern	■□□□ □□□□ □□□□ □□□□ □□□□	□■□□ □□□□ □□□□ □□□□ □□□□	□□■□ □□□□ □□□□ □□□□ □□□□	□□□■ □□□□ □□□□ □□□□ □□□□
Data	B0100	B0101	B0110	B0111
Pattern	□□□□ ■□□□ □□□□ □□□□ □□□□	□□□□ □■□□ □□□□ □□□□ □□□□	□□□□ □□■□ □□□□ □□□□ □□□□	□□□□ □□□■ □□□□ □□□□ □□□□
Data	B1000	B1001	B1010	B1011
Pattern	□□□□ □□□□ ■□□□ □□□□ □□□□	□□□□ □□□□ □■□□ □□□□ □□□□	□□□□ □□□□ □□■□ □□□□ □□□□	□□□□ □□□□ □□□■ □□□□ □□□□
Data	B1100	B1101	B1110	B1111
Pattern	□□□□ □□□□ □□□□ ■□□□ □□□□	□□□□ □□□□ □□□□ □■□□ □□□□	□□□□ □□□□ □□□□ □□■□ □□□□	□□□□ □□□□ □□□□ □□□■ □□□□

	Page B0111: Ones digit of input number			
Data	B0000	B0001	B0010	B0011
Pattern				
Data	B0100	B0101	B0110	B0111
Pattern				
Data	B1000	B1001	B1010	B1011
Pattern				
Data	B1100	B1101	B1110	B1111
Pattern				

	Page B0111: Tens digit of input number			
Data	B0000	B0001	B0010	B0011
Pattern	■■■■ ■□□■ ■□□■ ■□□■ ■■■■	■■■■ ■□□■ ■□□■ ■□□■ ■■■■	■■■■ ■□□■ ■□□■ ■□□■ ■■■■	■■■■ ■□□■ ■□□■ ■□□■ ■■■■
Data	B0100	B0101	B0110	B0111
Pattern	■■■■ ■□□■ ■□□■ ■□□■ ■■■■	■■■■ ■□□■ ■□□■ ■□□■ ■■■■	■■■■ ■□□■ ■□□■ ■□□■ ■■■■	■■■■ ■□□■ ■□□■ ■□□■ ■■■■
Data	B1000	B1001	B1010	B1011
Pattern	■■■■ ■□□■ ■□□■ ■□□■ ■■■■	■■■■ ■□□■ ■□□■ ■□□■ ■■■■	□□■□ □□■□ □□■□ □□■□ □□■□	□□■□ □□■□ □□■□ □□■□ □□■□
Data	B1100	B1101	B1110	B1111
Pattern	□□■□ □□■□ □□■□ □□■□ □□■□	□□■□ □□■□ □□■□ □□■□ □□■□	□□■□ □□■□ □□■□ □□■□ □□■□	□□■□ □□■□ □□■□ □□■□ □□■□

Page B1111 contains a self-diagnostic program, which tests every pixel, row and column individually.

Appendix B: Solutions

Here are the solutions to the tasks from the different chapters in this book.

Question 1		Solution
?	What is the decimal value of 1111?	8+4+2+1=15
	What is the decimal value of 0110?	4+2=6
	What is the decimal value of 1010?	8+2=10
	What is the binary value of decimal 15?	1111
	What is the binary value of decimal 12?	1100
	What is the binary value of decimal 9?	1001
	How can you easily spot an odd binary number?	Odd numbers always end with a 1. (and even numbers with a 0).

Question 2		Solution
?	Set the variable to any value, let’s say 10 (B1010). Then, disconnect the USB power from the Binary Counter. Will the Variable remember its value when you reconnect the USB power?	Yes, the Variable module remembers its value as long as the knob does not get turned in either direction.

Question 3		Solution
?	What is the biggest number that the variable can produce? What is its binary value?	The biggest number it can produce is 15, which is B1111.
	What is the smallest number that the variable can produce? What is its binary value?	The smallest number is 0, which is B0000.
	Set the variable to 15 (that's B1111) and then turn the knob to the right for one notch. What is the output of the variable? Can you explain it?	The output will be B0000, this is the lower 4 bits of 16, which, in binary, is B10000. But because our variable can only see 4 bits, it ignores the fifth bit. So our Variable believes that the result is B0000 only.

Question 4		Solution
?	What is 0101 + 1010?	1111 (5+10=15)
	What is 0010+0010?	0100 (2+2=4)
	What is 0111+0001?	1000 (7+1=8)
	What is 1111 + 0001? Why are all the Adder's LEDs off?	10000 (16). This is a 5 bit number. All LEDs are off because the Adder can only work with 4 bits. It is simply 'blind' to the 5th bit.

Question 5	Use your B4 to compute:	Solution
?	10+3+2	13+2=15
	1+1+1	2+1=3
	2+3+8	5+8=13
	3+2+1	5+1=6
	0+0+1	0+1=1

Question 6	Try to find	Solution
?	A happy smiley	Page: B0011, Data: B0011
	A question mark	Page: B0011, Data B0001
	A dot	Page: B0011, Data B1100
	A closing bracket	Page: B0011, Data B1101
	The number 8	Page: B0010, Data B1000

Question 7		Solution
?	Experiment a bit further by changing the value of the Variable to B0000, B0001, B0002, etc. What other interesting animations can you produce? Have fun!	This is an open ended experiment. There is no right or wrong answer

Appendix C: Quick Reference Guide

Decimal	Binary	Pink LED	Yellow LED	Green LED	Blue LED
0	0000	○	○	○	○
1	0001	○	○	○	●
2	0010	○	○	●	○
3	0011	○	○	●	●
4	0100	○	●	○	○
5	0101	○	●	○	●
6	0110	○	●	●	○
7	0111	○	●	●	●
8	1000	●	○	○	○
9	1001	●	○	○	●
10	1010	●	○	●	○
11	1011	●	○	●	●
12	1100	●	●	○	○
13	1101	●	●	○	●
14	1110	●	●	●	○
15	1111	●	●	●	●