MARTY THE ROBOT

Educator's Guide

Everything you need to get started

Contents

Introduction to Marty the Robot

Marty is a fully-programmable, WiFi-enabled, walking robot who helps you teach programming, electronics, and mechanical engineering in a fun and engaging way. By promoting creativity and learning Marty acts as an entryway into computer science but is also adaptable enough to be used up to University level.

There is no need to be a programming expert to use Marty in a classroom. Your Marty class pack comes with initial setup support and there are lots of tools online to help you get up and running.

Pupils can also lead their own development and learn as they go through a Build – Play - Learn ideology. If bought in kit-form, Marty offers a systematic and logical build process with pre-set constraints, there are also simple guided tasks suggested in each language Marty operates in. For more advanced mechanical design you can also 3D print new parts for Marty, either from our website or bespoke designs.

Marty can be programmed in languages including Scratch, Python and Javascript. From each of the programming environments you can control whole movements or individual motors, read sensors, and make your own tasks for Marty to do. For more advanced programming and robotics you can upgrade Marty with a Raspberry Pi computer and run ROS, the Robot Operating System as used in academia.

Marty the Robot is specifically designed to help young people develop STEM skills. This includes:

- Computational thinking and valuable programming skills using real languages;
- Electronics, mechanics and engineering design;
- Sequencing, planning and organisation skills;
- Collaboration, leadership and peer interaction skills;
- Creativity, inventive thinking and initiative;
- Critical thinking through exploratory analysis;
- Logical thinking in an engaging and creatively fulfilling environment;
- Evaluation and problem-solving skills;
- Opportunities to engage with a global community through ICT.

By exposing young people to real world problems, Marty can encourage kids to think and act like budding engineers, scientists and designers.

Program Marty with Scratch

Getting comfortable with Marty

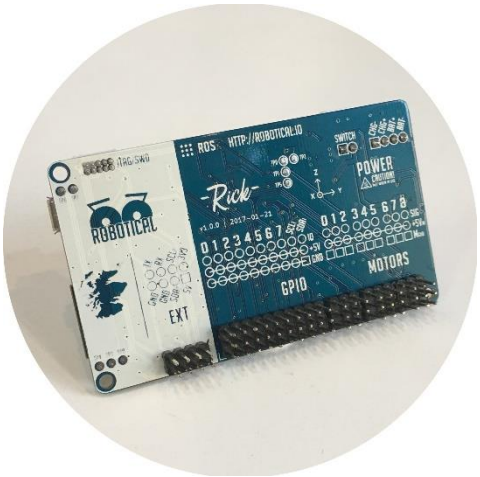
Real robots are full of highly advanced technology, and Marty is no different. Although the technology may be complex, that doesn't mean it has to be scary. Marty offers advanced computing and engineering but makes it approachable for people of all ages. With big eyes, moving eyebrows, and a funky way of walking, Marty is designed to add character, personality, and fun to the process of learning

Let's explain a bit about how Marty works. Some of the most important parts are the:

Control Board (the "Brain")

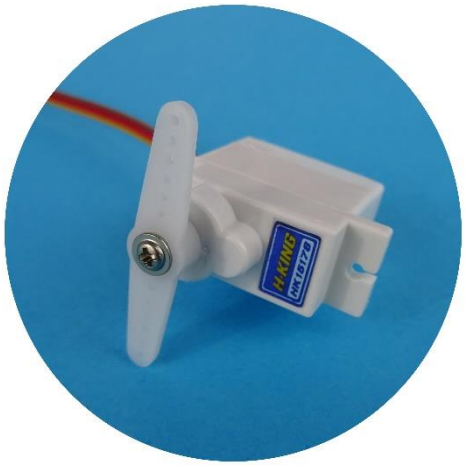
The Control PCB (Printed Circuit Board) is where all the complex triggers and reactions happen once you program Marty. Think of the PCB as Marty's brain. It's the central system for all things Marty does and feels.

On the control board there is a network of components that help Marty move and sense. For example, the control board has an accelerometer that tells Marty when he's not standing up and how he's moving. It's like our own human balance system, which tells us when which way is up and can make us feel dizzy.



The control board also speaks to Marty's motors and communicates what the sensors are reading. Again, this is similar to our own nervous system, which receives signals from our senses, for example when we bump into something, and tells us how to react, in this example to stop walking and say "ouch". You can use Marty's brain to send signals to the servo motors and read the sensors (like the tilt sensors, or the bump switches you can attach to his feet) to allow Marty to react physically when he feels something – like shaking hands if he feels somebody grab his arm.

The more sensors you add, the more Marty will be able to interact with the world around him, all of which is possible due to his control board brain. The control board acts as the portal through which we communicate with Marty, and Marty can communicate with the world.

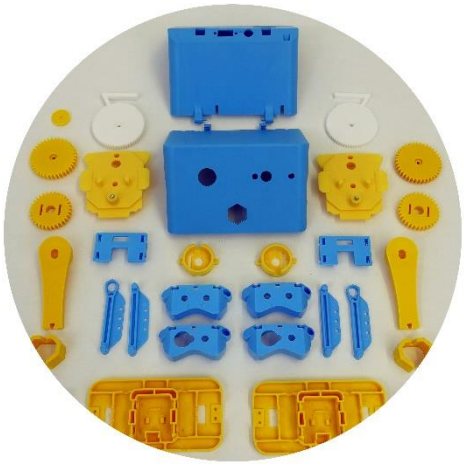


Servo Motors ("Muscles")

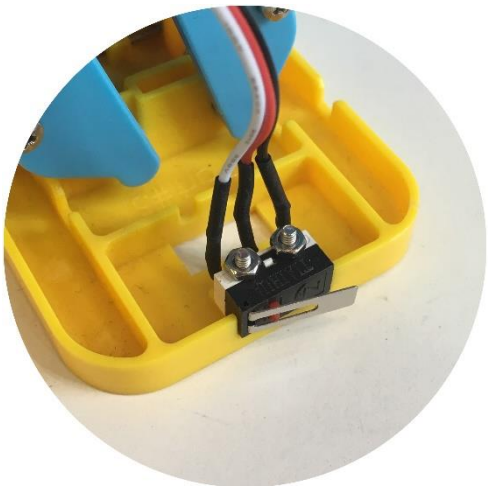
Marty has nine servo motors which allow him to move – three in each leg, one for each arm, and one to move the eyebrows. The servos have position sensors in them, which allow them to move to specific commanded positions – so if you command Marty to move his motors through a series of poses, they know what to do.

One of the unique things about Marty is his walking mechanism, which uses fewer motors than a traditional biped. That makes Marty easier to use, and more energy efficient.

Plastic components (“Body”)



Marty has 55 plastic parts that make up most of his body. These parts form the skeleton of Marty, as well as a protective shell around Marty’s vital components. The body parts fit together using stainless steel nuts and bolts, making it possible to assemble and disassemble Marty many times. As well as mechanics, linkages, and physics, the components lay a foundation for lessons surrounding material science and manufacturing processes.



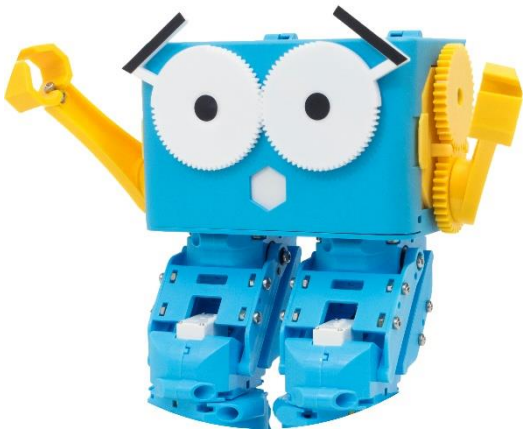
Sensors

Any real robot needs to interact with the world around it. Industrial robots used to just perform the same movement again and again, but these days we expect robots to be able to react to real life situations. Built in sensors for Marty include force sensing on all the joints except the eyebrows, tilt and acceleration sensing, and the kit comes with two bump switches you can attach to the feet. They can be used to detect collisions, or facing down to detect ledges and keep Marty from falling off things!

You can also add extra sensors, for example distance sensors which you could use to help Marty solve a maze or react when somebody sits down in front of him. For more advanced users it’s possible to add a Raspberry Pi computer on Marty, and then give Marty a microphone, or a camera to let him detect faces and play football autonomously!

Gears & independently controlled motors (“Personality”)

Another special trait of Marty is his unique personality and expressiveness. Rather than having a screen for a face, Marty has big eyes with moving eyebrows. Emotion by motion is possible because each of Marty’s limbs are controlled by their own separate motors. Marty’s personality helps young people engage with him and reduces barriers for those who may not be as comfortable when engaging with technology. Personality is the key to helping a wide range of individuals to be comfortable when engaging with technology and taking that first step to a potential future career. Through this Marty offers an approachable entry into STEM engagement.



Marty and the classroom

Marty is an adaptable tool that can be used across both primary, secondary, and even tertiary education. By sparking the excitement of young people and harnessing their infectious enthusiasm, Marty can rapidly promote STEM learning and engagement. In some schools computing science is in danger of becoming a mundane functional experience. Marty brings the excitement back by offering a tangible interactive experience and promoting a learn-by-play ethos. By exposing young people to real world problems, they can begin to think and act like budding engineers, scientists and designers. This will help them to secure a place in the ever-developing world of work.

Primary (KS1 & KS2/Early & Second Level)

The primary stage development with Marty introduces foundation concepts such as sequencing, storytelling and the relationship between instruction and operation. It supports the abilities to:

- Identify patterns in everyday tasks;
- Understand and structure information;
- Understand, predict and problem solve block-based programming languages;
- Understand and communicate using technical language;
- Develop a sequence of instructions to align to a design brief.

Secondary (KS3 & KS4/Third - Senior Level)

The secondary stage development with Marty explores more advanced concepts such as analysis, computational thinking and text-based programming. It supports the ability to:

- Analyse algorithms for correctness and identify faults;
- Understand the world through computational thinking;
- Design testing and build computing solutions;
- Introduce object-oriented programming concepts;
- Understand data structures and process real world data from sensors

Learn about mechanical systems during the Marty build

Marty to student ratio

Within a standard co-operative environment, we recommend no more than three students per Marty. However, there are opportunities for students to work on a one to one individual basis for personal projects, or larger collaborative groups for class projects.

Building your Marty

The recommended format for building your Marty is detailed in our documentation. It is recommended that you assemble all the separate component parts and then assemble them together to finish the build, starting with the **Legs,** progressing to the **Arms** and then onto the **Head.** Once all these components are assembled, you can attach them together and add the electronics.

It is possible to build Marty in groups as the Arms, Legs and Head can all be constructed in parallel, with the final assembly and installation of the electronic components being completed as a group activity. We recommend the following groupings:

Assembly Step	Build Steps	Individual	Group of 2		Group of 3		
		Student 1	Student 1	Student 2	Student 1	Student 2	Student 3
Leg 1	1-11	X	X			X	
Leg 2	1-11	X		X			X
Arm 1	12-14	X	X		X		
Arm 2	12-14	X		X	X		
Head Base	15-17	X		X	X		
Head Face	18-19	X	X		X		
Final Assembly	20,24,25	X	X	X	X	X	X
Electronics	21-23	X	X	X	X	X	X

Groups of 4+: Building in groups of 4+ is not recommended as it often leads to a lack of ownership which results in an increase of mistakes during the build. However, given adequate supervision and direction, larger groups can be successful and should be left to the individual educator's discretion.

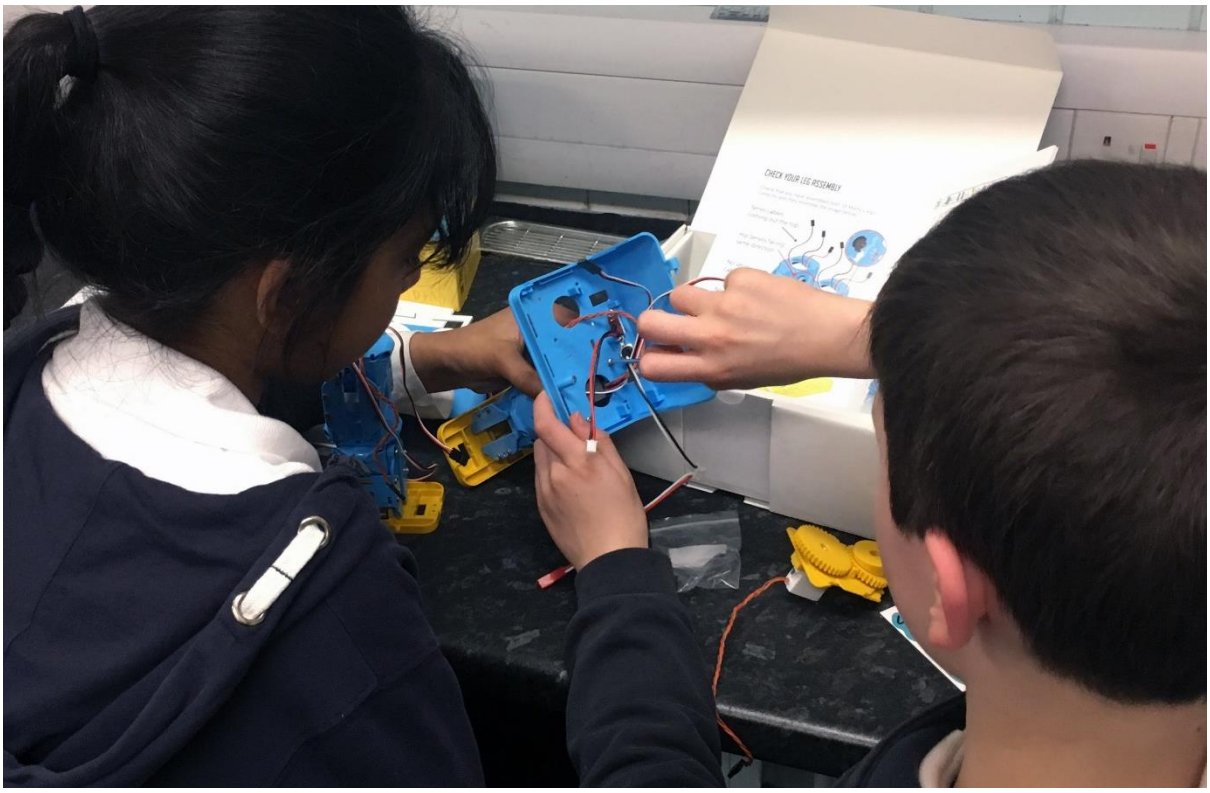
Storage tips

- Marty's original packaging can be repurposed as a safe storage container. The double wall cardboard acts to keep Marty safe and secure whilst also being stackable and compact.
- For those that require easy Marty transport between classrooms we suggest a wheeled trolley (not supplied). This can allow for a quick and organised set up in a variety of different classroom settings.

Classroom advice

We have put Marty through rigorous safety testing. Please read the following points to help secure that safety in the classroom environment.

- Marty is not a toy - adult supervision is required.
- Marty contains small parts which may present a choking hazard.
- Do not expose Marty to extreme heat or a naked flame.
- Avoid excessive physical shock or vibration.
- Do not immerse Marty or any of Marty's components in water and do not use in an environment where there is a possibility of getting Marty wet.
- Never leave Marty charging unattended. Do not charge Marty for more than 24 hours.
- Only use Marty with the supplied rechargeable battery.
- Do not disassemble, deform, puncture, short circuit, modify in any way or heat the battery. Improper use can result in fire causing serious injury and property damage.
- Stop using and disconnect the battery immediately if the battery begins to bulge or becomes hot.
- Battery must be charged with the supplied charger only. Never use a modified or damaged charger
- The charging cable is not a toy.
- Connect charging cable to a USB 5v 1.5 A DC supply.
- If connecting to a transformer, only use with a transformer for toys. The transformer is not a toy.
- Do not clean Marty with liquid.



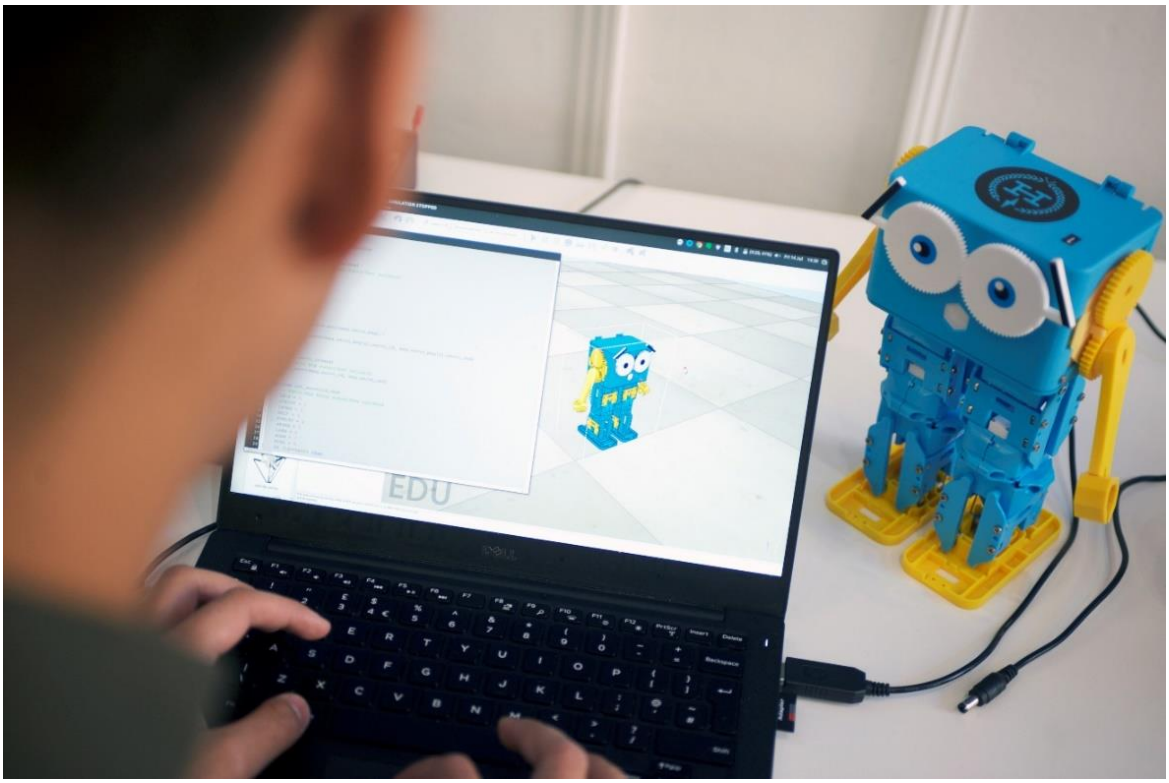
Working in teams helps to support co-operation

Marty in the curriculum

Marty acts as a valuable tool to engage young people in their education, no matter which national or regional curriculum you use. Some core technology learning skills are listed below:

Curriculum for Excellence (Scotland) *(Technologies Experiences and Outcomes, 2016)*	**National Curriculum (England / Wales / N.I.)** *(The national curriculum in England framework, 2013)*
• Knowledge and understanding of key concepts in the technologies. • Curiosity, exploration and problem-solving skills. • Planning and organisation skills. • Creativity and innovation. • Skills in collaborating, leading and interacting with others • Critical thinking through exploration and discovery. • Making connections between specialist skills developed within learning and skills for work.	• Understand and apply fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation. • Analyse problems in computational terms and have repeated practical experience of writing computer programs in order to solve such problems. • Evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems.

Marty has the capacity to complement a young person's learning by hitting each of these targets in an engaging and adaptable manner. Marty lesson plans can be crafted to accommodate a wide-reaching range of curriculums and can include subject crossover content that expands the learning possibilities across disciplines.



Simulate Marty using VREP

The Marty learning process

Marty has been developed to accommodate learner progression. Beginners can program Marty to carry out simple sequences of instructions and react to simple inputs, intermediate learners can start to use real text-based programming languages and implement complex algorithms on Marty, and advanced learners can use real world robotics tools like ROS and explore subjects like kinematics and signal processing.

Learning with Marty

To get started with Marty we recommend you and your pupils just get stuck in and start to experiment! Recipe-based learning is a thing of the past and instead learning-by-play has presented itself as a style that can empower each pupil to direct their own learning journey.

The value of learning-by-play activities is evidenced by the freedom the student has to create things that are meaningful to themselves. These situations allow the student to approach the task with a sense of interest and care that can be missed in other learning styles. The result being that the student gains a deeper connection with the task and field of study, encouraging their interest in the subject.

Once you've started to get to know Marty we recommend asking the following set of questions to help you use Marty in the classroom:

- What problems did you encounter and how did you solve them?
- Will your students be able to overcome these problems?
- How much of an introduction will Marty require and how will you structure it?
- What areas of Marty use will need the most encouragement?
- How will you link Marty to the relevant benchmarks and learning checkpoints?
- How much time will you need to allocate for each lesson introduction?
- How can you make sure Marty is an efficient and effective learning tool for all?

Problem solving with Marty

Learning Progression

<table>
<tr><th>Hardware Skills</th><th>Software Skills</th><th>Software Progression</th><th>Age</th></tr>
<tr><td>• Explore mechanisms</td><td>• Use logical reasoning to predict program behaviours
• Explore and comment on logical processes using computational thinking</td><td rowspan="2">• Marty the Robot app controller
• Scratch</td><td>4-8 years</td></tr>
<tr><td>• Recognise materials
• Discuss and create engineering design solutions
• Explore electronic systems</td><td>• Explain programming language concepts using technical terms
• Create computing solutions in response to a design brief
• Identify errors and debug</td><td>8-11 years</td></tr>
<tr><td>• Design and construct assemblies
• Explore and explain engineering design solutions
• Product technology analysis
• Understand and suggest sustainability improvements</td><td>• Select the appropriate tool to fulfil a selected design solution
• Understand the fundamentals of hardware and software communication.</td><td>• Python
• JavaScript</td><td>11-14 years</td></tr>
<tr><td>• Discuss engineering principles and their impact
• Analyse material performance
• Explore sensors
• Understand and create electronic circuits</td><td>• Compare algorithms for correctness and efficiency
• Select the appropriate tool to fulfil a high-level computing solution with reasoning justified</td><td rowspan="2">• C++
• ROS (Robotic Operating System)
• V-Rep simulation</td><td>14 – 18 years</td></tr>
<tr><td>• Explore mechatronics
• Understand robotic principles and standards
• Kinematics</td><td>• Use ROS and other advanced tools
• Advanced algorithms and data processing – image processing, speech, filtering etc.</td><td>18+</td></tr>
</table>

Coding Marty

Getting started with Scratch

Scratch is the simplest way to begin programming with Marty. By combining code blocks you can build basic to complex programs to bring your Marty to life. Scratch has been designed to help people take that first step into programming by teaching basic sequencing and linear progression.

Getting started with Marty Scratch: Robotical have developed a Scratch extension in the form of a series of Scratch blocks to be used specifically with Marty. This version of Scratch can be accessed in two ways depending if you are working online or offline with a Robotical Hub.

Online: with access to the internet, go to robotical.io/getstarted and scroll to the ***"Get Started with Scratch"*** section. Within this section, you can follow our guide to familiarise yourself with the program. Pressing the ***"Marty ScratchX"*** button will take you to the Robotical Marty Scratch extension.

Offline with a Robotical Hub: If you have access to a Robotical Hub you can simply connect to its "Robotical Hub" WiFi hotspot and then type 172.24.1.1 (the Hubs IP address), this will bring up a menu that allows you to select several extensions for Scratch.

Developing with Python

Python helps you take the first step away from block-based coding and into textual programming.

Getting started with MartyPy: MartyPy is a Python library for Marty. Libraries are just collections of code that we've written that get all the complicated stuff out of the way, making it easier to use and work with. For example, MartyPy contains the definition of the "walk" function, which can be used to make Marty walk. For information on how to get started with Python, go to Robotical.io/getstarted and select the "Get Started with Python" tutorial.

Advanced progression with Raspberry Pi

Raspberry Pi is a small in-expensive computer that can be optionally paired with Marty to allow for the addition of further sensors and functionality.

How to get started with Raspberry PI: Robotical have a pre-built image for a Raspberry Pi available with ROS and a bunch of other programming tools already installed to use with Marty. You can download it in the Raspberry Pi section of the Robotical Website (Robotical.io). You can then follow the instructions that will lead you through the configuration, networking and installation of you Pi.

Further learning

There are several ways in which you can go further with Marty. One of the recommended ways is to integrate with the Robotic Operating System (ROS). ROS is a set of advanced robotics tools that are used widely within academia and industry to manage many of the services you would expect from the OS in a computer system. With ROS you can program Marty to functional autonomously and perform tasks without user intervention.

Support

Growing outside the classroom

If you or your pupils would like to learn more about programming, we recommend the following sites to find specialist tutorials. Each of these sites offers open-source learning for whoever requires them - no matter the age or development stage.

Scratch - https://scratch.mit.edu

Python - https://wiki.python.org or https://www.codecademy.com/

JaveScript- https://javascript.info/ or https://www.codecademy.com/

Raspberry Pi - https://www.raspberrypi.org

C++ - http://www.cplusplus.com

ROS - http://wiki.ros.org

Robotical EduWiki

Robotical hosts an Educational Wiki where educators can upload and share their experience of Marty in the classroom. This platform acts to support those that are new to Marty as well as to allow a system of co-education through shared lesson plans. In addition to this there are software specific community sites that host challenges and content specific to there platform. Such platforms include ScratchEd, MicroBit and Raspberry Pi.

Connecting education to community

As an open-source tech tool, Marty has gathered a community following. This community consists of makers that are actively contributing additional content whenever they can. The result is a wide network of challenges and activities that can be shaped for the classroom.

Contact us

Education

Finlay Page
finlay@robotical.io

Distribution & Sales

Myles Bax
myles@robotical.io

Support

support@robotical.io