



Science in School

The European journal for science teachers

ISSUE 74 – September 2025

Topics Astronomy/space | Coding | Engineering | Physics | STEAM

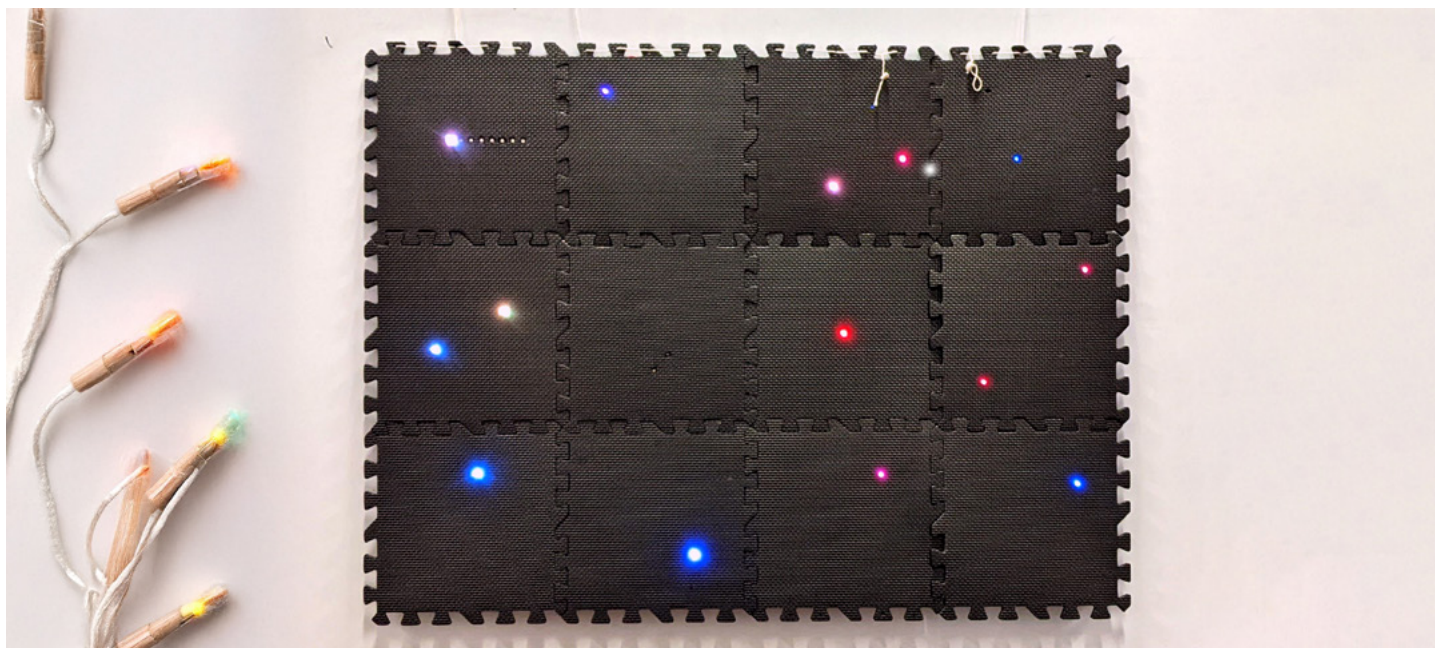


Image courtesy of the author, image editing Christina Hof/Science in School

Wall of stars: illuminate stellar life cycles with physics and coding

Jed Marshall

Written in the stars: use microcontrollers and LEDs to model stellar life cycles, scaling billions of years into minutes while exploring stellar evolution.

Stars undergo fascinating evolutionary processes over millions or billions of years, but these vast timescales can be difficult for students to grasp. This project offers a hands-on approach that makes stellar evolution tangible and engaging by using microcontrollers like BBC micro:bits and red-green-blue (RGB) light-emitting diodes (LEDs) to create proportionally scaled models of stellar life cycles.

Students will translate astronomical data into programmable light sequences, visually representing the changing colours and relative durations of each stellar phase. The project emphasizes computational thinking, problem-solving, and collaborative learning, while reinforcing key astrophysics concepts. It serves as an excellent complement to tools like the stellar evolution software Window to the Stars.

Learning Objectives: identifying stellar life-cycle stages, understanding vastly different stellar timescales, applying proportional scaling, developing microcontroller programming skills, and collaborating to solve modelling challenges.

The principles of using a microcontroller to change the colour profile of an RGB LED can be used with any common microcontroller used in schools. Here, we use the BBC micro:bit; however, the principles will also work with Arduino, Raspberry Pi, or Crumble microcontrollers.

If you have not used a micro:bit before, there are plenty of beginner tutorials on the [BBC micro:bit](https://www.bbc.com/microbit) website.

These activities are suitable for students aged 11–9 (adaptable based on coding experience).

Activity 1: Understanding stellar life cycles and timeline challenges

Before students can model stellar life cycles, they need to understand what they're modelling and why the timescales present such an interesting challenge.

This activity takes 30–40 minutes.

Materials

- Teaching materials on stellar evolution
- [Stellar life cycle table](https://www.bbc.com/microbit)
- Writing materials and a calculator

Procedure

1. Recap stellar-evolution basics, adjusting depth for prior knowledge. Cover star formation, main-sequence burning, and mass-dependent end points.
2. Introduce the project goal: using micro:bits and LEDs to create proportional models of stellar life cycles, where LED colours represent evolutionary stages.
3. Organize students into groups of two or three. Half research Sun-like stars (~1 solar mass), and half research high-mass stars (~20 solar masses).



Safety notes

Electronic components should be handled with care. Ensure alligator clips do not cause short circuits by touching each other on micro:bit pins, and check all connections are secure before powering on. Supervision by the teacher is recommended for younger students or those new to electronics.



Image courtesy of the author

4. Groups research stellar timescales, that is, how long the stars spend at each stage, presented in millions or billions of years:
 - a. Sun-like stars: protostar (50 million years), main sequence (10 billion years), red giant (1 billion years), and white dwarf formation (100 million years).
 - b. High-mass stars: protostar (0.5 million years), main sequence (7 million years), red supergiant (1 million years), and supernova (instantaneous).
5. Introduce the scaling challenge: compress these timescales into a 10-minute model. We turn these timescales into a shorter time so that we can 'play out' the full evolution of the star using the micro:bit. Students can change the length of their programme later, but 10 min is a good starting point.
6. Students create and populate a simple table that maps stellar events from several billion years into 10 min.

Star stage	Stellar time	10-min model	Star colour
Protostar	1 million years	5 s (would be shorter but important to see)	Dim red
Main sequence	10 billion years	9 min	White/yellow
Red giant	1 billion years	55 s	Red/orange
White dwarf	A very long time	Choose a time before the model starts again	Dim white/yellow

Activity 2: Hardware setup and LED-colour exploration

Students become familiar with the hardware and learn how RGB LEDs can represent the colours of a star in its different stages and tell the story of a star by changing the colour of the RGB LED using micro:bit.

Students can write this code using JavaScript (traditional written code) or MakeCode blocks (graphical block-based programming), depending on their coding experience. Coding blocks are shown here to help guide students, and a JavaScript [code example for colour mixing](#) and is given in the supporting material.

Students or teachers unfamiliar with jumper cables and using prototyping boards can research their basic function. A detailed article [explaining prototyping boards and jumper cables](#) can be found online.

This activity takes 30–40 minutes.

Materials

Per group of 2–3 students

- 1 × BBC micro:bit or other microcontroller
- Computer with internet access for Computer with internet access for [MakeCode editor](#)^[1]
- 1 × USB cable (for programming and power)
- 1 × RGB LED for micro:bit (recommend Monk Makes RGB LED with integrated resistors^[2])
- 3 × 470 Ω resistors (if not using the Monk Makes version)
- 4 × jumper cables or alligator clips

Procedure

1. Students should connect the RGB LED to the micro:bit using jumper cables. Connect the ground to GND, red to P0, green to P1, and blue to P2 (via resistors if needed).
2. Test basic LED controls using the MakeCode editor. Start with a simple red light.

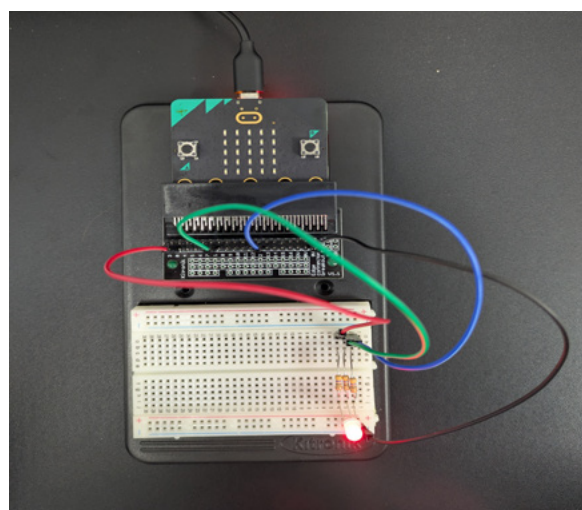
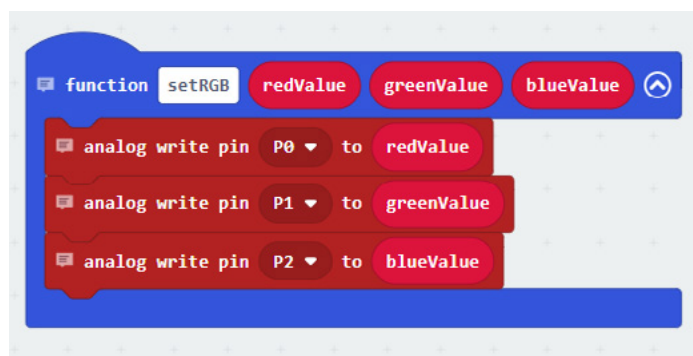


Image courtesy of the author

3. Students should use their knowledge of light and colour mixing, along with some trial and error, to experiment with different colours. This can be done by writing analogue values between 0 (no brightness) and 1023 (maximum brightness) to pins P0, P1, and P2, representing red, blue, and green, respectively.
4. Students should explore stellar colours by creating RGB combinations representing different stellar types, noting down the RGB values in the table from Activity 1:
 - a. Cool red stars (protostars, red giants): high red, low green/blue
 - b. Yellow-white stars (Sun-like): balanced warm colours
 - c. Blue-white stars (hot main sequence): high blue/green, moderate red
 - d. White stars (white dwarfs): balanced RGB values
5. To make the handling of the colour settings more comfortable, create a reusable function in, for example, JavaScript or block code.



Screenshot taken from Microsoft MakeCode

Image courtesy of the author

Activity 3: Programming stellar life cycles

Students combine their timeline understanding with coding skills to create functioning stellar models. Stars don't live forever, and their various life stages have vastly different durations. A Sun-like star might spend billions of years in its stable main-sequence phase, while its later stages are much shorter. High-mass stars live even faster, more dramatic lives.

Students use the same materials as in Activity 2 and can write the code using JavaScript (traditional written code) or MakeCode blocks (graphical block-based programming), depending on their coding experience. Coding blocks are shown below and a JavaScript [code example for high mass stars](#) is given in the supporting material. [Full example code](#) (in both forms) with comments is also provided.

This activity takes 45–60 minutes.

Procedure

1. As a class, decide on a total runtime for the stellar life-cycle model. Good starting points are 30 s (for quick testing) or 10 min (600 s, for a more detailed display).
2. In the next steps, students need to calculate appropriate times for the different life stages of the two star types. Each group should do this for their assigned star type; the example shown below is for a solar-mass star.
3. How many real years should 1 s of the model represent? This is the scaling factor. For example, if the Sun's main sequence is 10 billion years and we want it to last 5 min (300 s) in the 10-minute model, what would the scaling factor be?

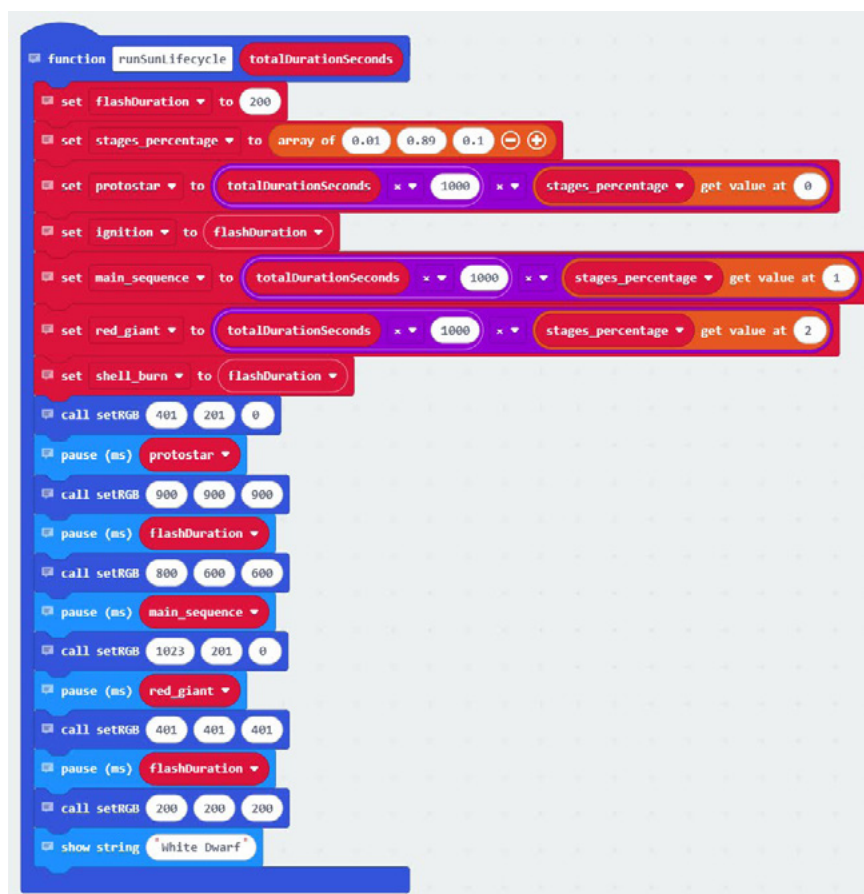
Challenge question: if 1 billion years of a star's life is represented by 30 s in the model, how long would a 50-million-year protostar phase be?

4. Some events, like the Sun's ignition flash or a supernova, are astronomically very brief. In the model, it makes sense to give these a fixed, short duration (e.g., 0.2–0.5 s), so they are noticeable.
5. If a stage is very short in reality (like the Sun's protostar phase relative to its main sequence), its scaled time might become too brief to see on the LED. Decide on a minimum observable duration for any nonflash stage (e.g., 1–5% of the total time) such as 1–5 s. While this is not 'realistic', it still allows us to see every stage and important event for the star.

Tip:

it is useful to use seconds for the function input, which will decide how long the simulation runs; however, calculations will be done in microseconds. Convert seconds into microseconds by multiply by 1000.

6. The table created in Activity 1 was for 10 min; however, we might want to use different timescales. Use percentages and the input runtime to determine the length of each stellar stage. This information will be stored in an array (a programming term for a table; see the annotated [example code](#)). 'Short' events don't need to be included, as these would be impossibly short to represent on the scale. We want to represent each stage of the stellar life cycle as a percentage of the overall time (e.g., perhaps 91% or 0.91 for the main-sequence stage). The total life-cycle duration is multiplied by the percentage to get a value for how long this stage lasts.
7. Use mathematics to create time durations for each stage. Remember to use an appropriate short time (e.g., 0.2 s) for any 'flash' or short event, because if they were truly to scale, they would be imperceivable.
8. The pause function can be used to maintain the current RGB values for a given value of time. After setting the RGB values for a particular stage (e.g., protostar), we then want to pause for the correct length of time for that stage.
9. Add a trigger, such as button A being pressed, to call the function (to call a function means to use that function in the code) for the Sun's story. Remember the input is total runtime in seconds. We can adjust this to 10–30 s to trial the stages, but then we may wish to set it to 600 s for it to take 10 min.
10. Call the function and test the stars.



Screenshot taken from Microsoft MakeCode

Image courtesy of the author

Activity 4: Creating a classroom star wall

This part is optional, but it's a nice way to bring the project together and give the students the opportunity to display their work.

Transform individual models into a collaborative classroom display to demonstrate stellar diversity. Each group can be given a black interlocking foam floor tile, which can then be joined together at the end.

This activity takes 45–60 minutes.

Materials

- A star setup per group, as created in Activity 3
- Dark backing material (interconnecting foam floor tiles work well)
- Battery packs
- Mounting materials (e.g., Velcro tape or screws and spacers)

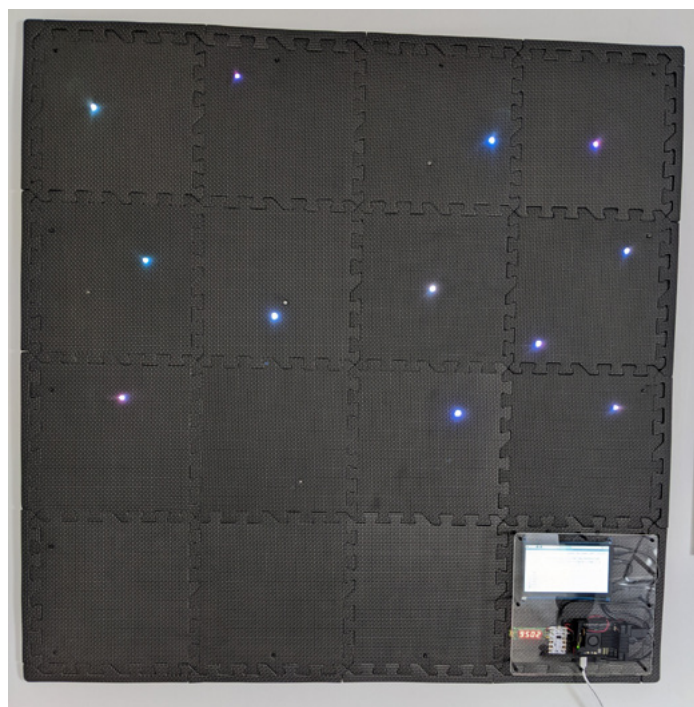
*Image courtesy of the author*

1. Each group is given a tile or section of the display.
2. Select a random position within this tile to mount the star. The micro:bit and battery pack can be mounted on the back of the tile; a small hole can be made with a sharp pencil to place the LED through.
3. The tiles can then be joined together to create a night sky/star field. Students can start their programmes at random times to demonstrate that our night sky is ever evolving and changing.

4. This will happen with our night sky, but over several billion years. The star Betelgeuse is due to supernova within the next 150 thousand years. Will civilizations in the future use the same constellations and asterisms?
5. One key feature that is not reflected by the model is that high-mass stars will have a much shorter lifetime than those that are Sun-like. This should be discussed, and as an optional extension, the model can be modified to reflect this.

Optional extensions

1. Modify the timescales of the models so that Sun-like stars and massive stars are to scale with each other. While the development of a Sun-like star takes an hour, massive stars reach the end of their lives after just a few seconds. After the death of a massive star, another massive star (the same LED) begins to shine after a random pause. This way it becomes apparent that massive stars pop up in galaxies like fireworks, while Sun-like stars need a long time for their evolution.
2. Students can invent constellations from the fantasy star field they have created. They can map the random positions of the stars onto a grid and try to create shapes, animals, or mythologies about the patterns they spot. This is also an opportunity to discuss the [cultural aspects of astronomy](#).



A final star display showing the interlocked foam tiles and randomly placed LEDs. A Raspberry Pi and a spare Crumble microcontroller are also included so users can programme their own star should they choose.

Image courtesy of the author

Discussion

This project provides a compelling and interactive method for students to engage with the vastness of stellar evolution. By translating abstract astronomical data into a tangible, light-based model, students not only reinforce their understanding of scientific concepts, but also develop crucial skills in programming, problem-solving, and collaborative design. The journey from a single LED to a classroom constellation can illuminate the wonders of the cosmos in a uniquely memorable way.

Students discover that main-sequence phases dominate stellar lifetimes, high-mass stars evolve dramatically faster than Sun-like stars, and scaling vast timescales requires careful mathematical consideration.

Key discussion points:

- Why do high-mass stars evolve faster? (Higher core temperatures drive faster nuclear burning.)
- How do stellar colours relate to temperature and evolutionary stage?
- What challenges do astronomers face when studying processes that take millions of years?
- How might stellar evolution affect planetary systems?

Acknowledgements

This project was developed with support from a Royal Society grant in partnership with Alexandra Park School, University of Surrey, and NEXT+ Vibenshus Gymnasium in Copenhagen. We thank the BBC micro:bit Foundation, MakeCode developers, and the Window to the Stars^[3] team for their educational resources. «

References

- [1] MakeCode for micro:bit: <https://makecode.microbit.org/>
- [2] Monk Makes RGB LED for micro:bit: https://monkmakes.com/mb_rgb
- [3] Graphical user interface, Window to the Stars, to the TWIN single and binary stellar-evolution code: <https://rizzard.surrey.ac.uk/window.html>

Resources

- Find the [full example code](#) with comments.
- Check out NASA's [educational resources on stellar evolution](#).
- Get inspired by further [teaching resources using the micro:bit](#).
- Try out educational computing resources from [Code Club](#) and [the Raspberry Pi Foundation](#).
- Read this clear and visually rich booklet on [the life cycle of stars](#).
- Take a journey through different 'sky cultures' to discover how people across time have reimagined the same stars into unique constellations.
- Learn the basics of block coding by using a micro:bit computer and create a timer for science experiments: Bowen GM, German S, Khan S (2023) [Introducing block coding: using the BBC micro:bit in the science classroom](#). *Science in School* **61**.
- Try some simple experiments to illustrate temporal additive colour mixing: Anta A, Goiri E (2024) [Colour magic: additive mixing and coloured shadows](#). *Science in School* **70**.
- Be amazed at how easy it is to estimate the Sun's temperature by using just some water, sunlight, and simple equipment: Kardaras I (2025): [Estimation of the Sun's temperature without leaving the school](#). *Science in School* **72**.
- Learn more about the Sun's spectrum and what the stars are made of: Ribeiro CI, Ahlgren O (2016) [What are stars made of?](#) *Science in School* **37**: 34–39.
- Use light bulbs to explain why stars shine in different colours: Ribeiro CI (2015) [Starlight inside a light bulb](#). *Science in School* **31**: 37–42.
- Get your students to use their smartphones for some hands-on astronomy: Rath G, Jeanjacquot P, Hayes E (2016) [Smart measurements of the heavens](#). *Science in School* **36**: 37–42.
- Measure distances to the stars like real astronomers with this classroom activity: Pössel M (2017) [Finding the scale of space](#). *Science in School* **40**: 13–18.
- Discover how astronomers study the past of the Milky Way and peer into its future: Forsberg R (2023) [Galactic Archaeology: how we study our home galaxy](#). *Science in School* **64**.
- Learn about how astronomers imaged the black hole at the centre of our galaxy: Forsberg R, Harris R, Vieser W (2023) [How global teamwork revealed the mystery at the heart of our galaxy](#). *Science in School* **62**.
- Find out how to distinguish between real and fake astronomical images: Muñoz Mateos JC (2024) [CSI Astronomy: learn how to spot fake astrophotography images](#). *Science in School* **69**.
- Learn what happens when a giant star dies: Székely P, Benedekfi Ö (2007) [Fusion in the Universe: when a giant star dies...](#) *Science in School* **6**: 64–68.
- Find out the environmental consequences of light pollution: Henshaw C (2022) [Too much of a good thing – the problem of light pollution](#). *Science in School* **56**.

CC-BY



Text released under the Creative Commons CC-BY license.
Images: please see individual descriptions