



THINKING ABOUT PROTOTYPING

UNIT II
Chapter 4



PROTOTYPING

- With the Internet of Things, we are always looking at building three things in parallel: the physical Thing; the electronics to make the Thing smart; and the Internet service that we'll connect to.
- Making a prototype first has many benefits.
- You will inevitably come across problems in your design that you need to change and iterate.
- Doing this with a single object is trivial compared to modifying hundreds or thousands of products.

PROTOTYPING

The prototype, therefore, is optimized for ease and speed of development and also the ability to change and modify it.

Many Internet of Things projects start with a prototyping microcontroller, connected by wires to components on a prototyping board, such as a “breadboard”, and housed in some kind of container.

You might substitute prototyping microcontrollers and wires with smaller chips on a printed circuit board (PCB), and pieces improvised out of 3D-printed plastic with ones commercially injection-moulded in their thousands



The background of the slide features a collection of colorful 3D printed parts, including a red 'A' shape, a purple part with a circular hole, a green 'C' shape, and various rectangular blocks in pink, yellow, and blue. A large, semi-circular green graphic is overlaid on the bottom half of the image, containing the title and list.

SKETCHING

- A hardware sketch is one of the foundations of technology design process.
- While sketching on paper helps designers to give a form to their ideas, **sketching in hardware** takes it one step further and allows them to explore possibilities and feasibilities of their designs in a more physical way.
- With this process, designers can assemble technological components quickly and openly, helping them with the experiential aspect of the design.
- It does require some kind of software support most of the time, but it is more about the physical interactions of the components than virtual simulations.



FAMILIARITY

- Another option to consider is familiarity.
- If you can already program in Python, for example, maybe picking a platform such as Raspberry Pi, which lets you write the code in a language you already know, would be better than having to learn Arduino from scratch.

COSTS VERSUS EASE OF PROTOTYPING

- Familiarity with a platform may be attractive in terms of ease of prototyping, it is also worth considering the relationship between the costs of a platform against the development effort.
- For the first prototype, the cost is probably not the most important issue: the smartphone or computer options are particularly convenient if you already have one available, at which point they are effectively zero-cost.
- An important factor to be aware of is that the hardware and programming choices you make will depend on your skill set.



A collection of colorful geometric shapes, including blocks and letters, in shades of red, purple, green, pink, and blue, arranged on a white surface. A large green semi-circle is overlaid on the bottom half of the image, containing the title and list.

PROTOTYPES AND PRODUCTION

- **CHANGING EMBEDDED PLATFORM**

- When you scale up, you may have to move to a different platform, for cost or size reasons.
- you may find that porting the code to a more restricted, cheaper, and smaller device will bring many challenges.
- If the first prototype you built on a PC, iPhone you may go about replicating that functionality on your final target.
- if you've used a constrained platform in prototyping, you may have to make choices and limitations in your code.
- Dynamic memory allocation on the Arduino may not be efficient for using strings or complex data structures.



PROTOTYPES AND PRODUCTION

- PHYSICAL PROTOTYPES AND MASS PERSONALISATION
 - The production techniques that you use for the physical side of your device won't translate directly to mass production.
 - While the technique might change— injection moulding in place of 3D printing, for example—in most cases, it won't change what is possible.
 - Digital fabrication tools can allow each item to be slightly different, letting you personalize each device in some way.

A background image showing various colorful wooden blocks and letters in shades of red, purple, green, pink, and blue, arranged on a white surface. A large green curved banner is overlaid on the bottom half of the image.

PROTOTYPES AND PRODUCTION

- CLIMBING INTO THE CLOUD
 - The server software is the easiest component to take from prototype into production.
 - If you are running on a cloud computing platform, such as Amazon Web Services, you can even have the service dynamically expand and contract, as demanded.

The background of the slide features a collection of colorful wooden blocks and letters in shades of red, purple, green, pink, and blue, arranged on a white surface. A large, semi-circular green shape is overlaid on the bottom half of the image, containing the text.

OPEN SOURCE VERSUS CLOSED SOURCE

- WHY CLOSED?

- Closed source software can be defined as proprietary software distributed under a licensing agreement to authorized users with private modification, copying, and republishing restrictions.
- The source code is not shared with the public for anyone to look at or change.
- Closed source is actually the sort of arrangement that you would expect from most businesses, protective of their product and keen to maintain control over their brand and the user experience offered to their customers.



OPEN SOURCE VERSUS CLOSED SOURCE

- WHY OPEN?

- Open source software (OSS) is distributed under a licensing agreement which allows computer code to be shared, viewed and modified by other users and organizations.
- open source software is available for the general public to use and modify from its original design free of charge.
- This means that the software is improved over time, but it can often take plenty of interesting twists and turns with all of that evolution and can change form and shape entirely.

A background of colorful wooden blocks in various shapes and colors (red, purple, green, pink, blue, yellow) scattered on a white surface. A large green semi-circular shape is overlaid on the bottom half of the image, containing the text.

Disadvantages of Open Source

- **1. Difficulty of use**
- **2. Compatibility issues**
- **3. Liabilities and warranties**
- **4. Hidden costs**

A background image showing a collection of colorful alphabet blocks in various colors like red, purple, green, pink, and blue, arranged on a white surface. A large green semi-circular shape is overlaid on the bottom half of the image, containing the text.

MIXING OPEN AND CLOSED SOURCE

- Open source and proprietary software development used to be competing strategies.
- Now software firms are experimenting with strategies that mix the two models.
- Software companies are taking a "best of both worlds" approach by creating products that use a combination of OS and proprietary software code.