Embedded
Online
Conference

Building a Simple Command-Line Interface (CLI)

Nathan Jones

www.embeddedonlineconference.com

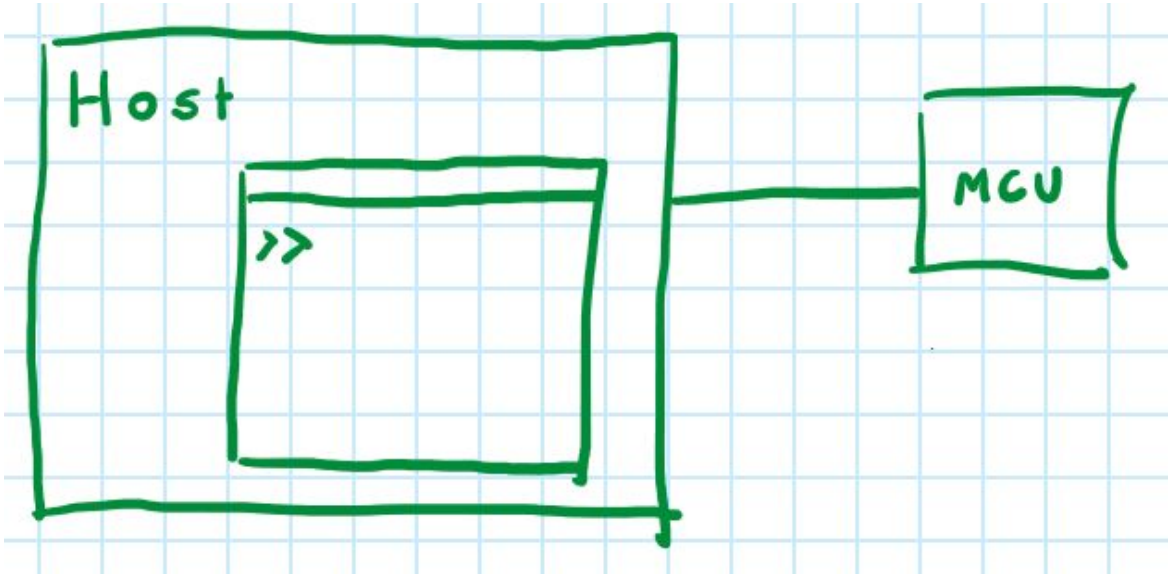
AGENDA

1. Why and What is it?
2. Plan of Attack
3. Step 1: Read from UART
4. Step 2: Simple commands
5. Step 3: Commands + values
6. Going Further

https://github.com/nathancharlesjones/simple-cli

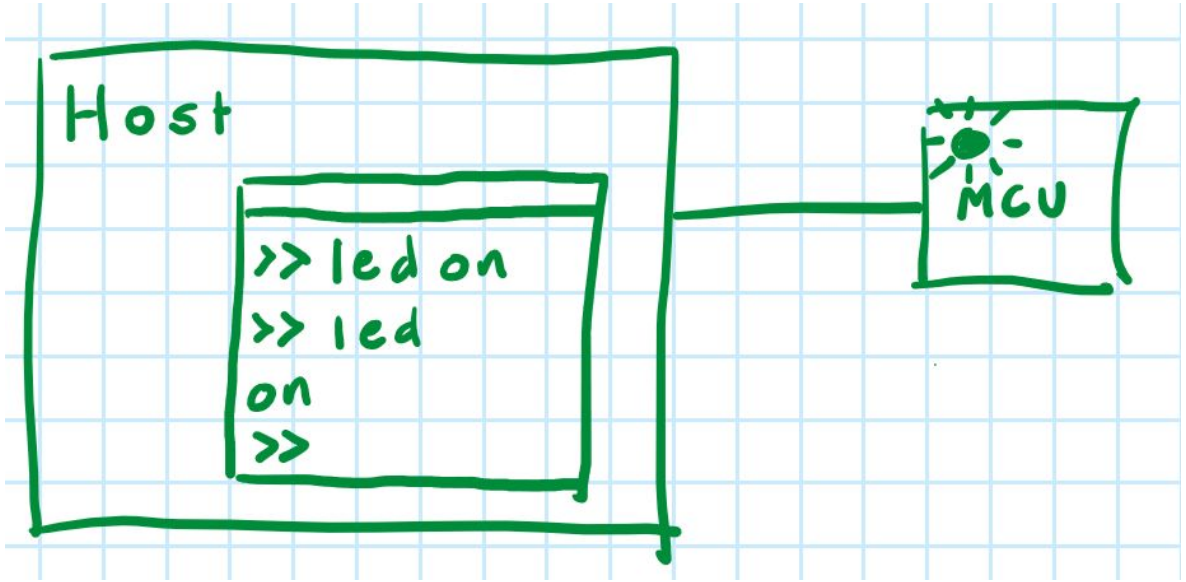
Why and What is it?

- Control tasks
- Query state
- REPL
- Load data/firmware
- On-chip debugging
- Monitor/small OS



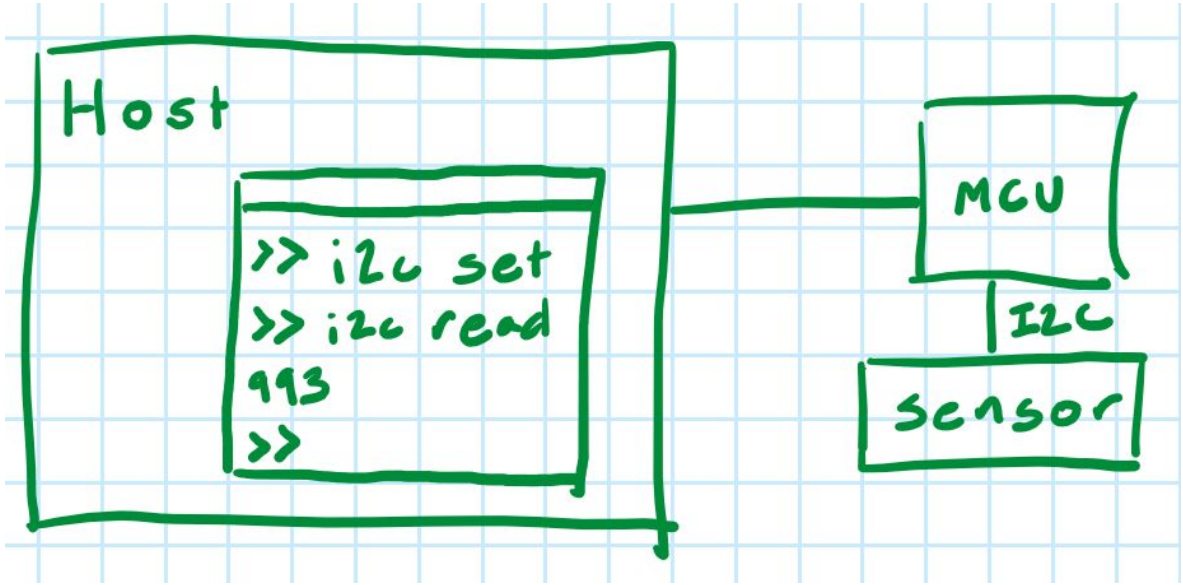
Why and What is it?

- **Control tasks**
- **Query state**
- REPL
- Load data/firmware
- On-chip debugging
- Monitor/small OS



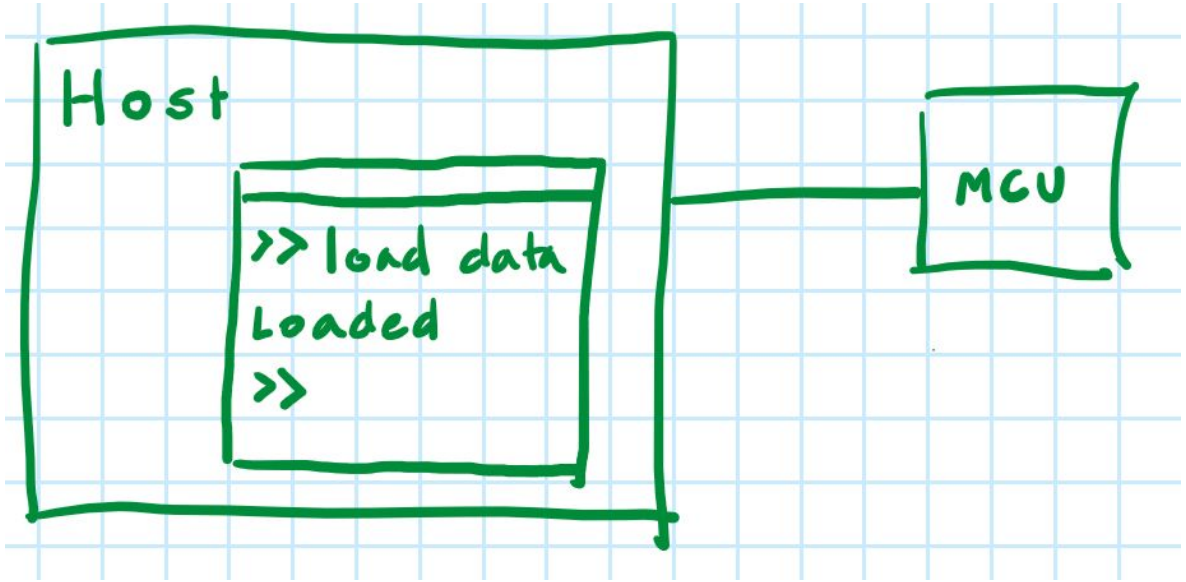
Why and What is it?

- Control tasks
- Query state
- **REPL**
- Load data/firmware
- On-chip debugging
- Monitor/small OS



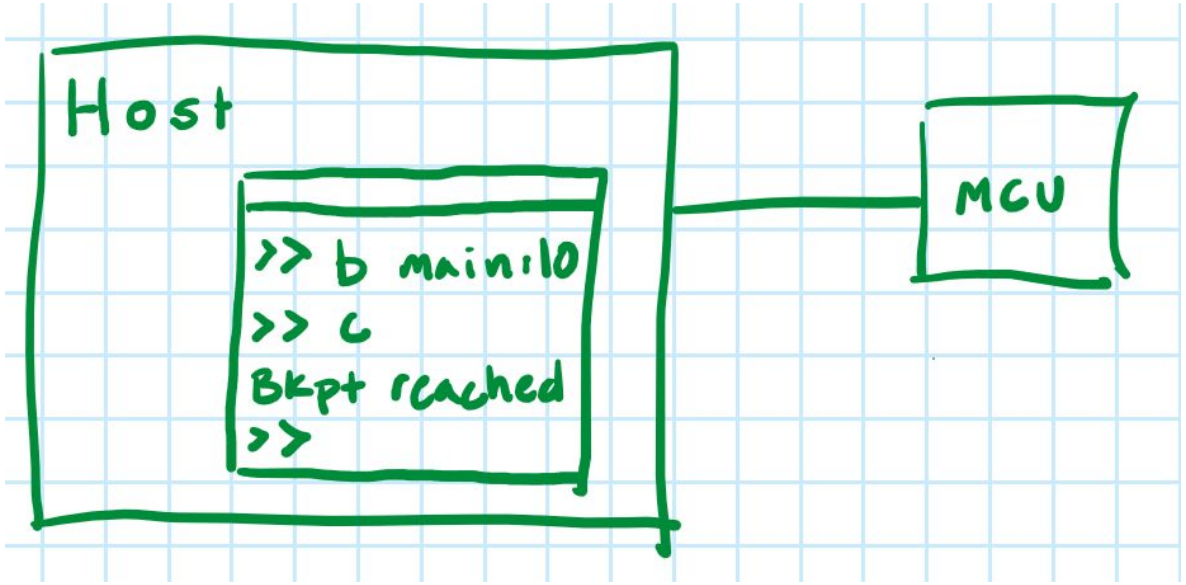
Why and What is it?

- Control tasks
- Query state
- REPL
- **Load data/firmware**
- On-chip debugging
- Monitor/small OS



Why and What is it?

- Control tasks
- Query state
- REPL
- Load data/firmware
- **On-chip debugging (Cortex-M)**
- Monitor/small OS



Why and What is it?

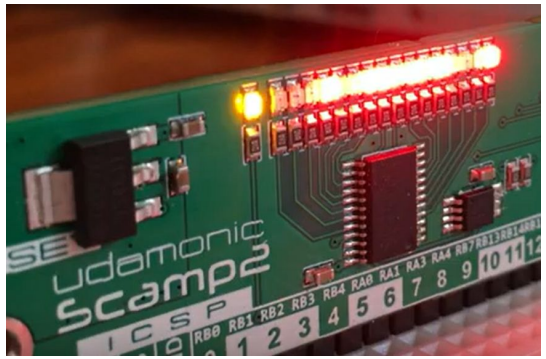
- Control tasks
- Query state
- REPL
- Load data/firmware
- On-chip debugging
- **Monitor/small OS**

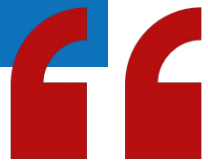
```
: blinken
  begin
     random leds
     blink
  key? until
  0 leds
;

ok


blinken
```

```
uShell 0.1.0
[host /]$ ls
d---- .
d---- bin/
d---- dev/
d---- etc/
-r--- readme.txt
[host /]$ cat readme.txt
Welcome to MicroShell DEMO implementation!
You will see how most common features work.
Enjoy!
[host /]$
```





The first thing I do in a new project is blink an LED. The next thing is to bring up a command-line shell. It's a great way to get stuff running quickly.

- andyturk (on the EEVBlog forum)

1 Command-Line Blinky

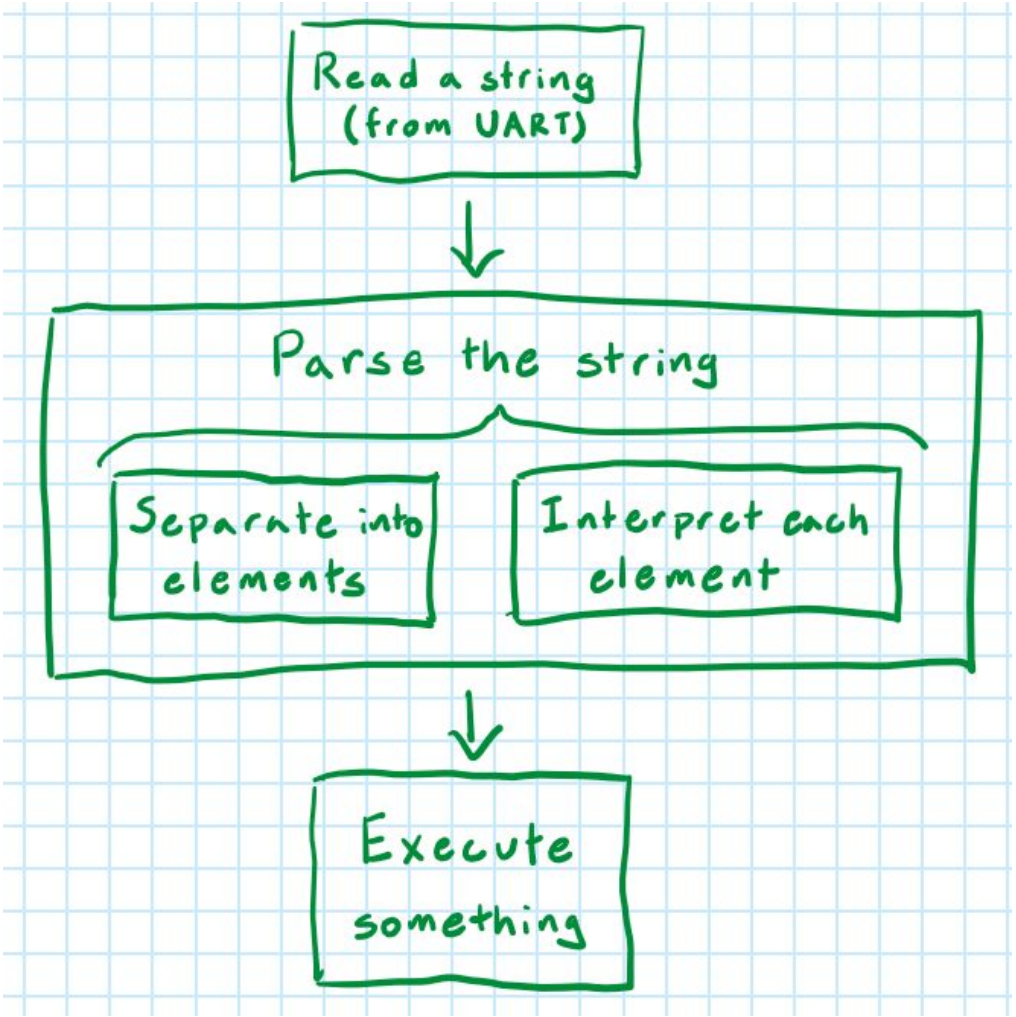
Message Dictionary	
on	Turns the LED on (at the last stored duty cycle & frequency)
off	Turns the LED off
dc <val>	Sets the duty cycle. Val is an integer percent value. Returns the current duty cycle if <val> is omitted.
freq <val>	Sets the blink frequency. Val is a float value in Hertz. Returns the current frequency if <val> is omitted.

1

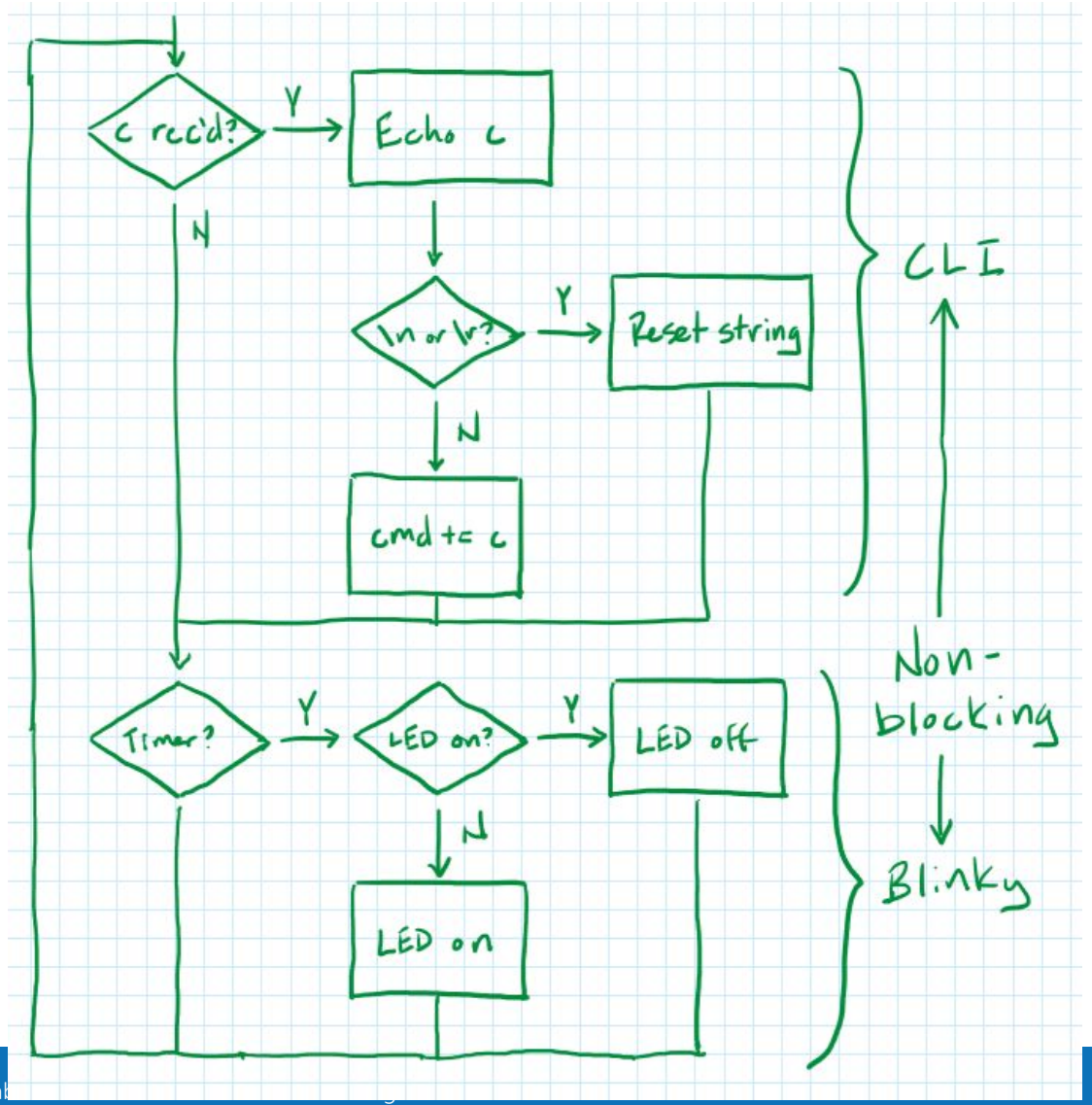
Live Display of ADC Values with PyQT

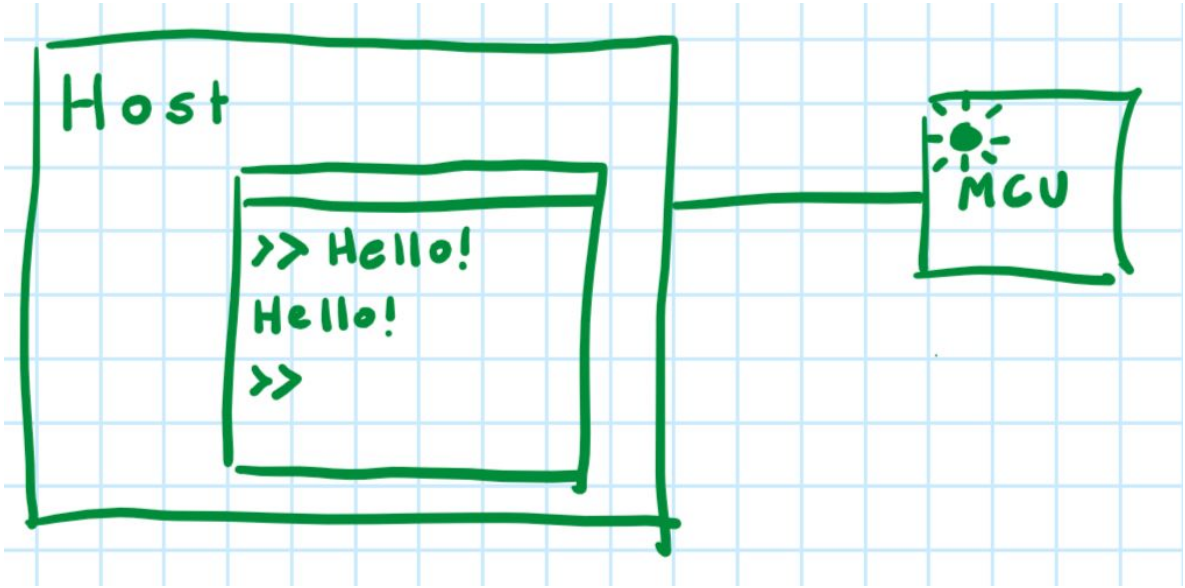
Message Dictionary		
To	r	Requests an ADC value
From	<val>	4-digit ADC value in ASCII

Plan of Attack



Step 1: Read from UART

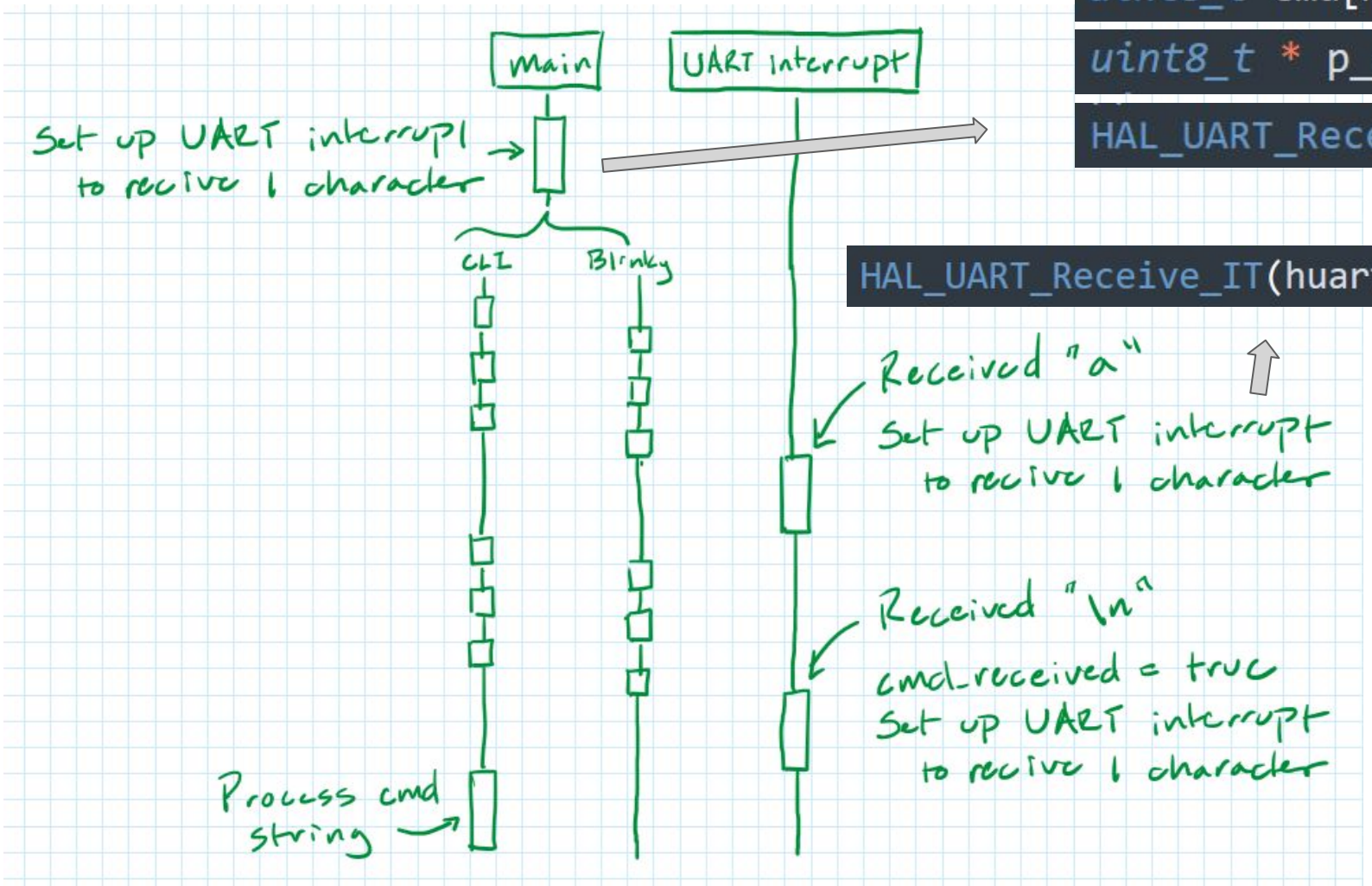




Step 1: Read from UART

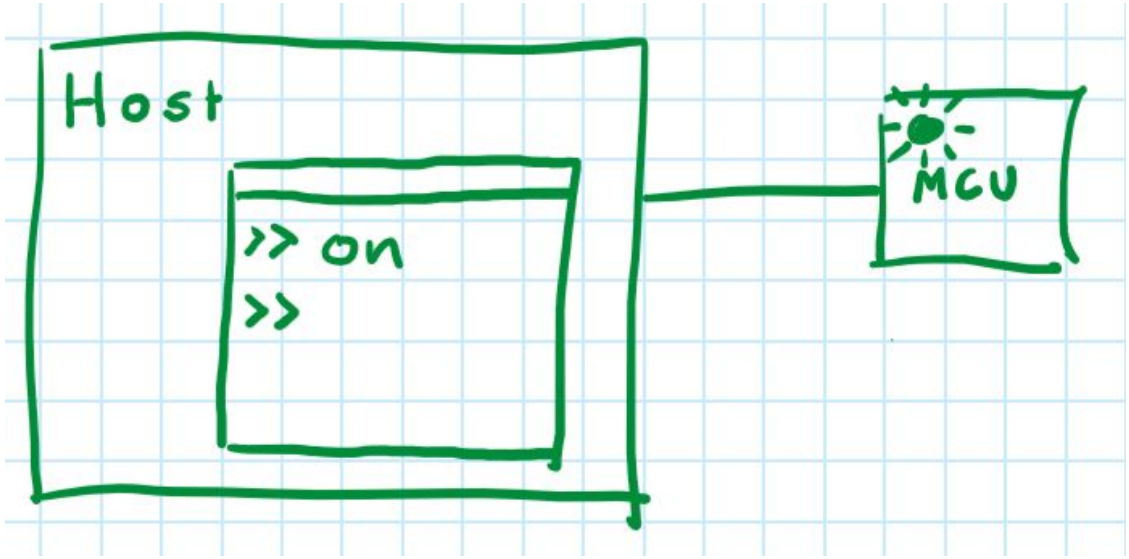
~~`Serial.available()`~~

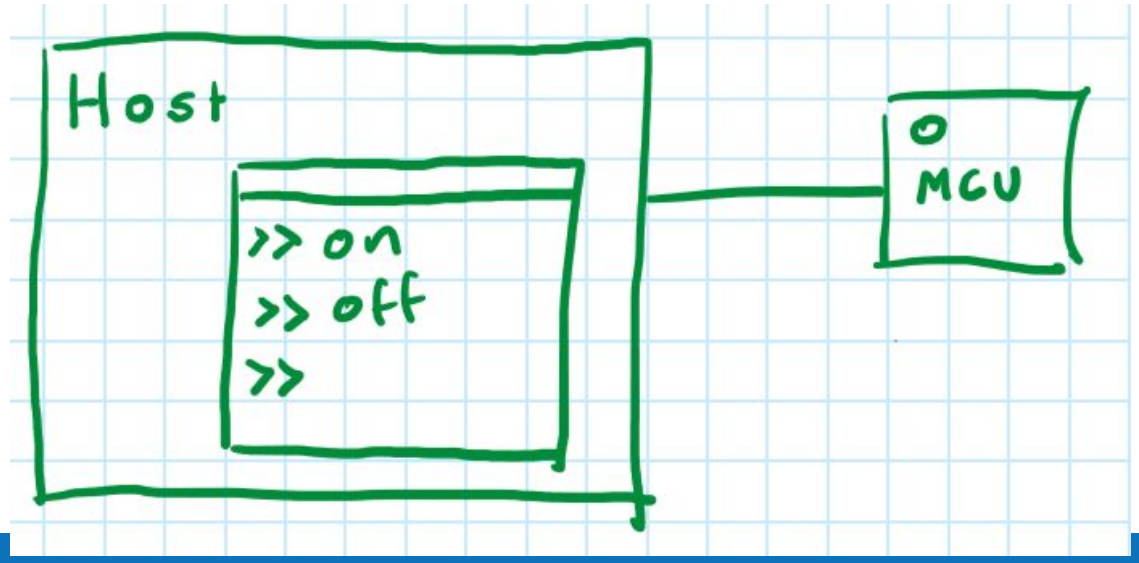
```
uint8_t cmd[MAX_CMD_LEN] = {0};
uint8_t * p_current_char = cmd;
HAL_UART_Receive_IT(&huart2, cmd, (size_t)1);
```




```
HAL_UART_Receive_IT(huart, ++p_current_char, (size_t)1);
```

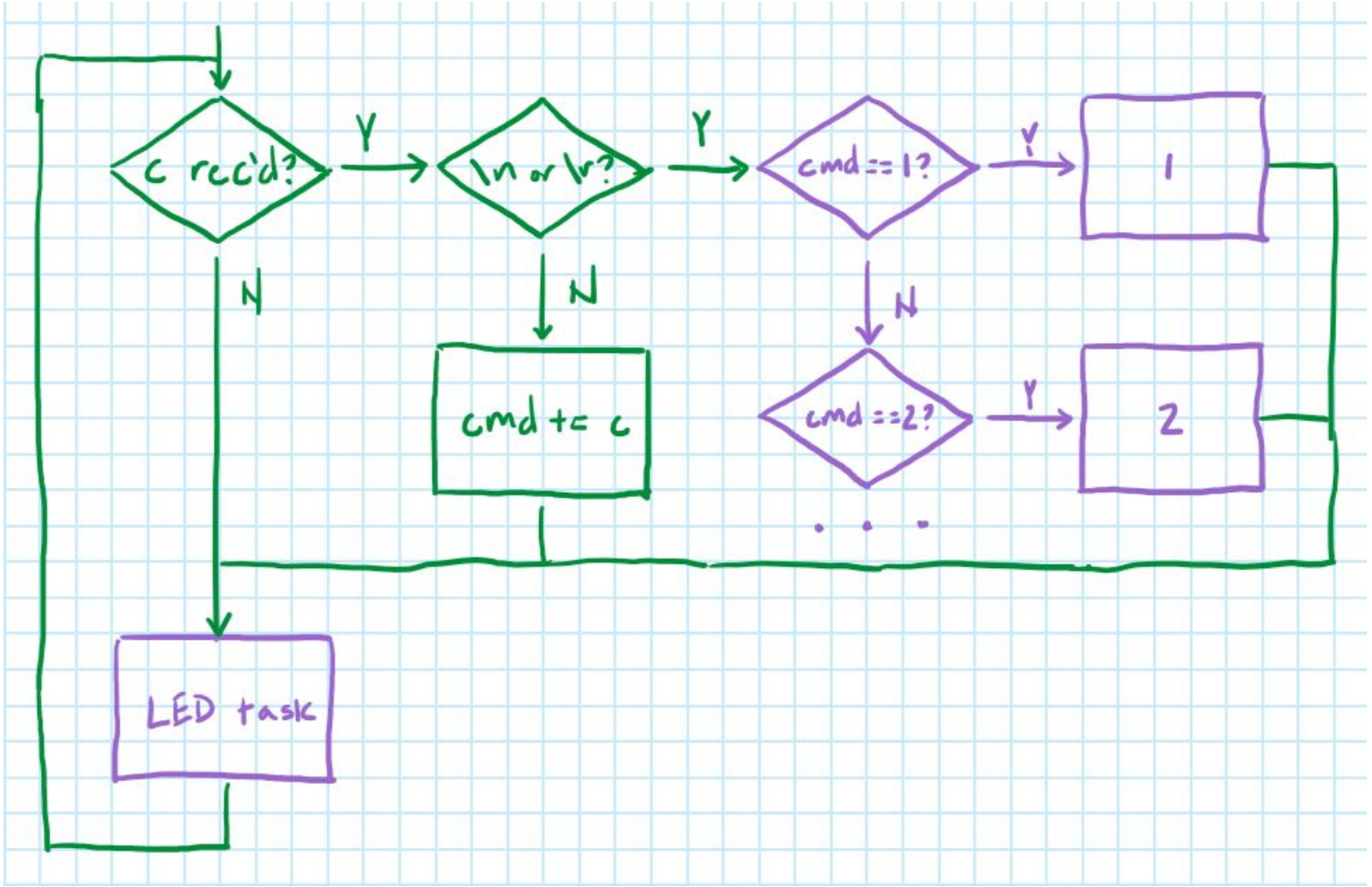
Step 2: Simple Commands





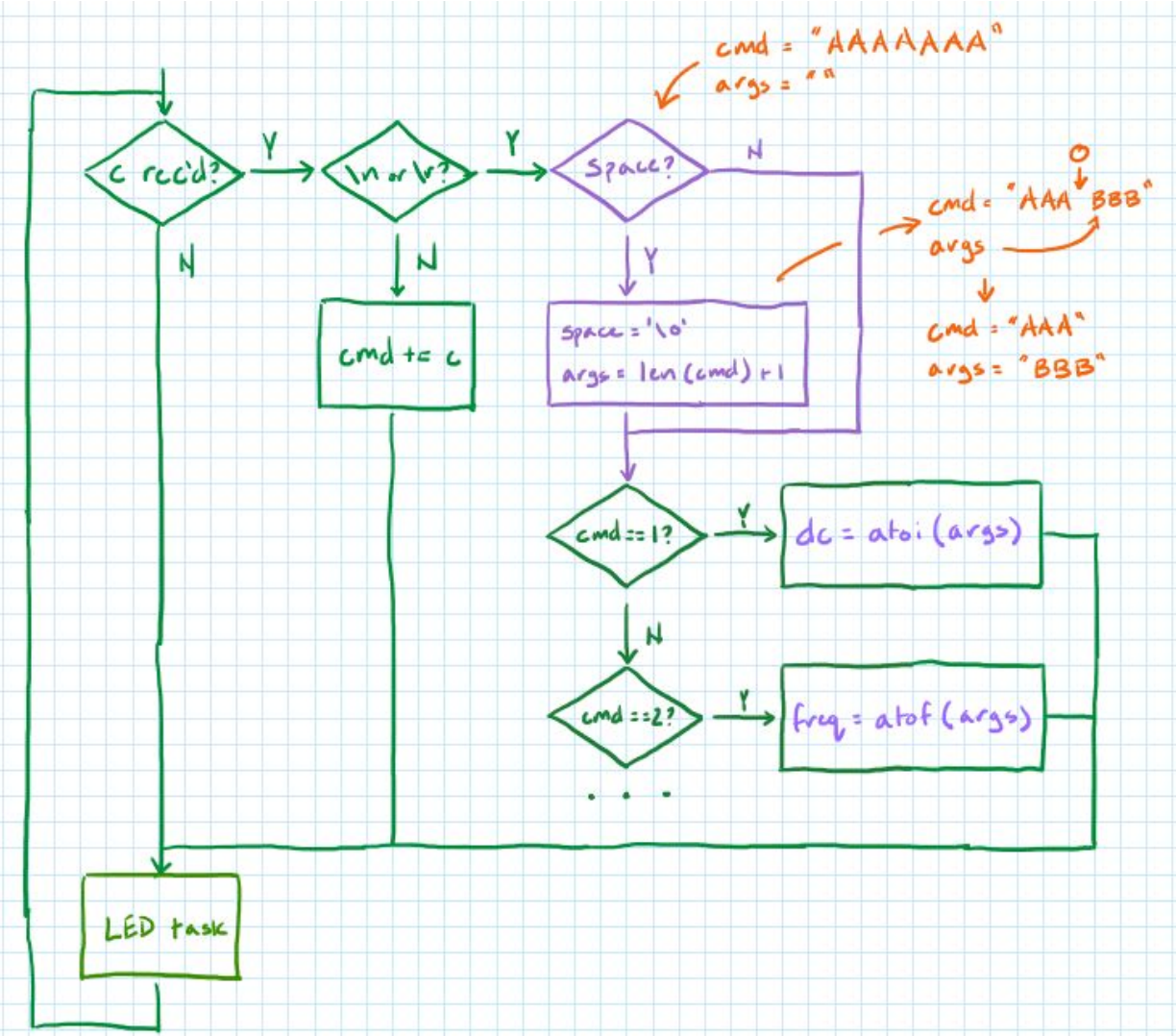
Message Dictionary	
on	Turns on the on-board LED
off	Turns off the on-board LED

Step 2: Simple Commands



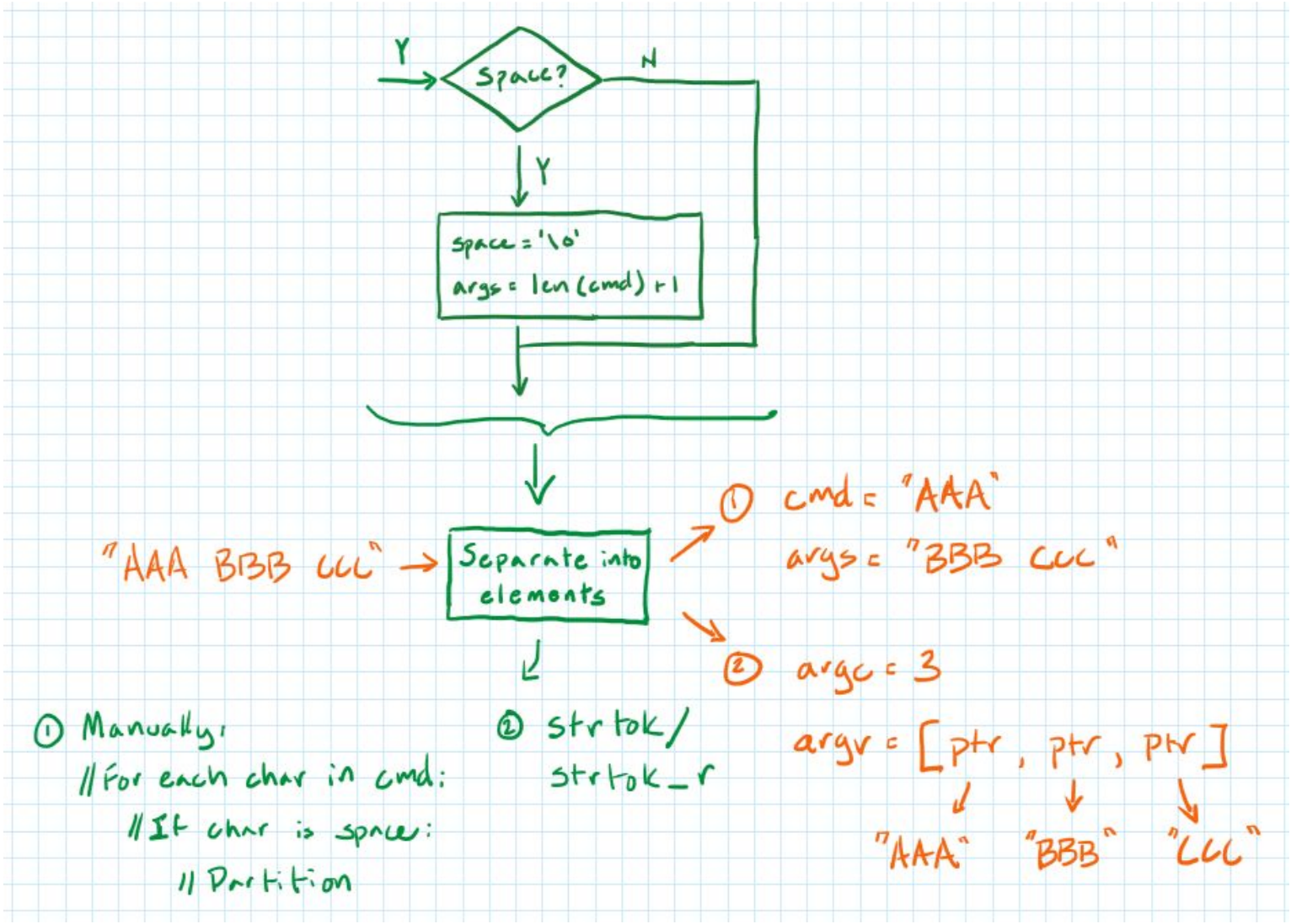
Message Dictionary	
on	Turns on the on-board LED
off	Turns off the on-board LED

Step 3: Commands + Values



Message Dictionary	
on	Turns the LED on (at the last stored duty cycle & frequency)
off	Turns the LED off
dc <val>	Sets the duty cycle. Val is an integer percent value. Returns the current duty cycle if <val> is omitted.
freq <val>	Sets the blink frequency. Val is a float value in Hertz. Returns the current frequency if <val> is omitted.

Step 3: Commands + Values



Going Further

CLI Libraries

- Anchor
- Memfault
- Args
- getopt / Gengetopt

Architectural Improvements

- Wireless communication
- Modules/Message passing
- RTOS
- Double-buffering
- Security
- Framing
- Error checking

6

Wireless CLI

Message Dictionary	
on	Turns the LED on (at the last stored duty cycle & frequency)
off	Turns the LED off
dc <val>	Sets the duty cycle. Val is an integer percent value. Returns the current duty cycle if <val> is omitted.
freq <val>	Sets the blink frequency. Val is a float value in Hertz. Returns the current frequency if <val> is omitted.

Going Further

Improving line reading

- getline
- linenoise

Interpreters/Monitors/OSes

- FlashForth
- uBASIC
- microshell
- RIDE Shell and C.impl interpreter (ELLO computer)

On-chip Debugging

THANK YOU

Embedded
Online
Conference

www.embeddedonlineconference.com