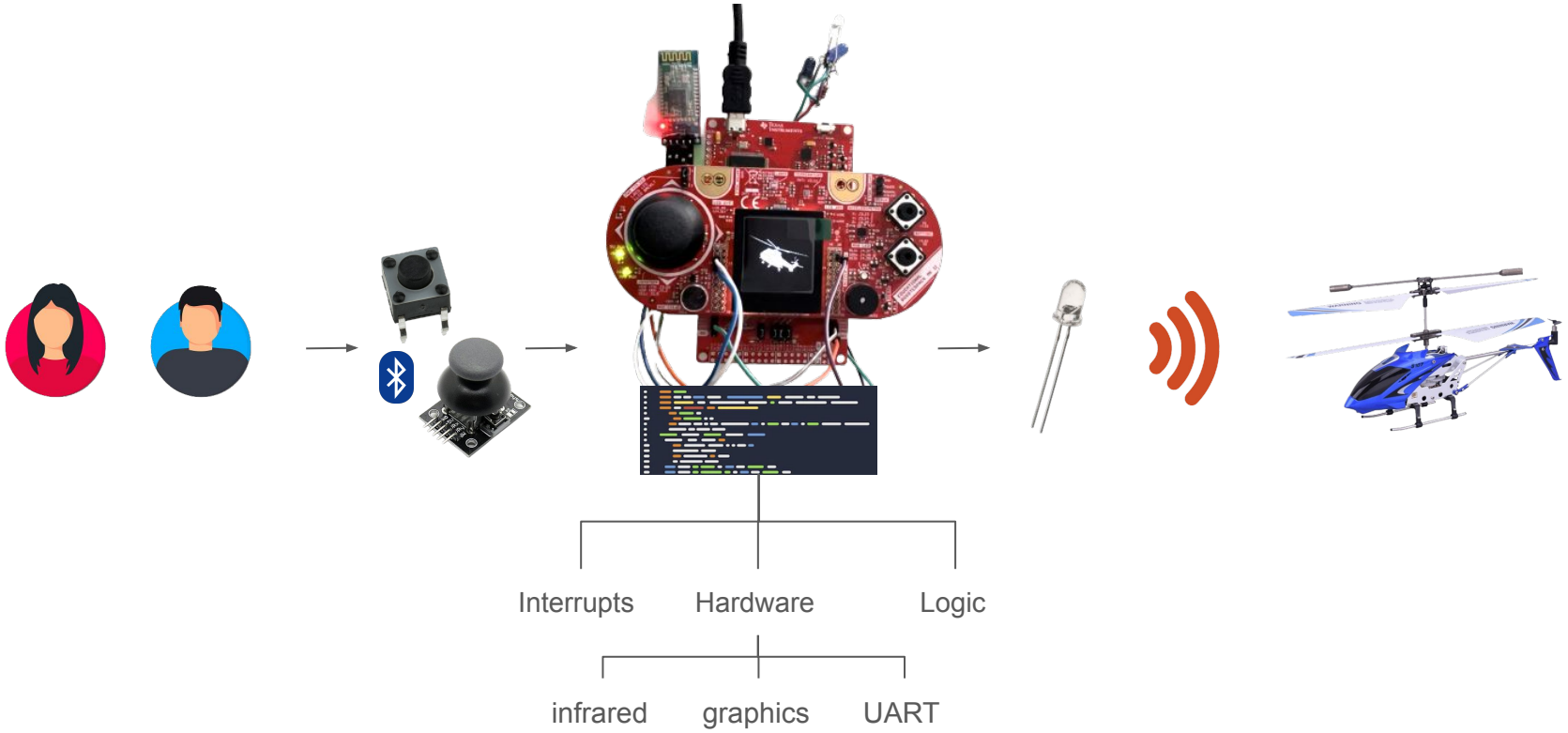
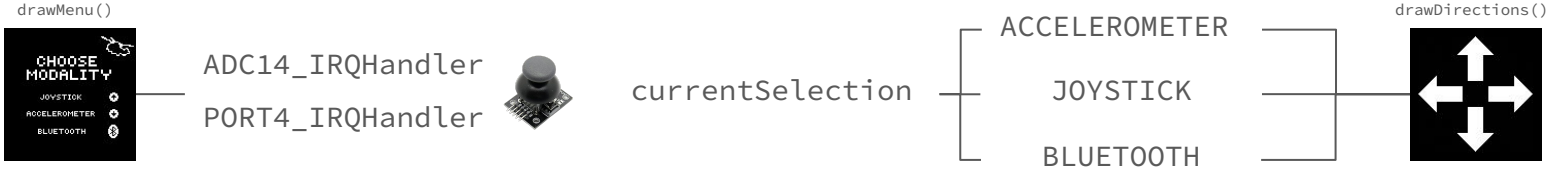
HELICOPTER CONTROLLER

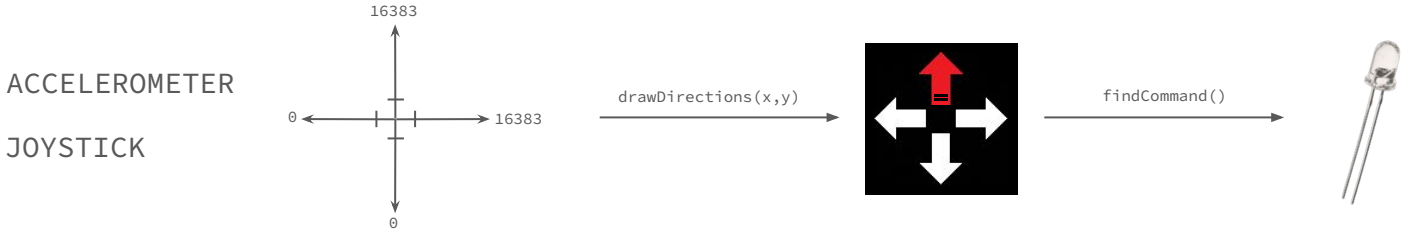
Implementation of an infrared controller for a helicopter using MSP432

System basic workflow



Software architecture





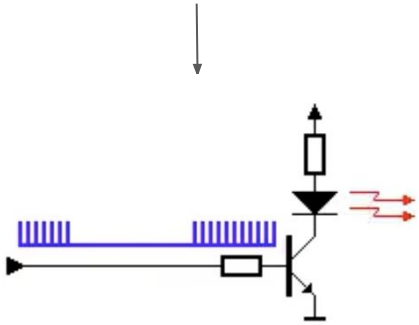
BLUETOOTH → UART initialization —customMSPgets()→ UART data exchange

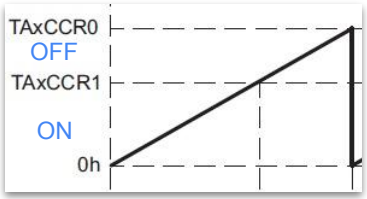
Code highlights

Infrared:

```
const uint16_t command[MAX_PULSE] =   {1950,250,400,250, ...}

const uint16_t command_p[MAX_PULSE] = {1950,200,750,200, ...}
```

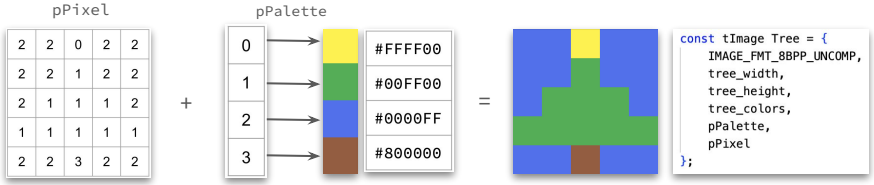




Graphics:

```
typedef struct {
    unsigned char BPP;
    unsigned int XSize;
    unsigned int YSize;
    unsigned int NumColors;
    const unsigned long * pPalette;
    const unsigned char * pPixel;
} tImage;
```

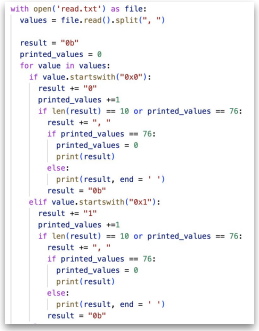
Defined in `<grlib.h>`



For two-colors images pixels can be represented with 0 and 1 (black and white) and the full image can be written as a sequence of strings like these: 0b00010111, 0b00110101 …

From png to RRRGGGBB (8bpp):
https://notisrac.github.io/FileToCArray/

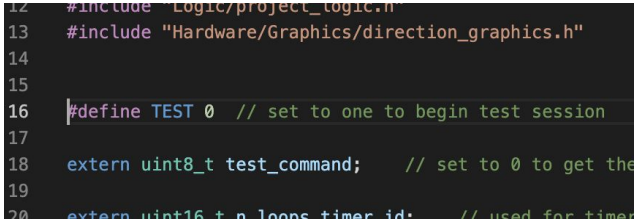
From 8bpp to 1bpp: this simple python script



Testing

`MSP432_Project/Logic/test.h`

- command sender testing
- timers testing
- graphics testing
- Bluetooth testing



Improvements

- Improve the IR transmitter range by providing more current through a transistor
- Add voice control through TinyML voice pattern recognition
- Add an accelerometer module and a basic communication system to send data to the MSP432 board and compute (and display) the altitude and speed data of the helicopter.