Bat Machine

Ultrasonic Sensor and Servo Controll With a FPGA

Lourdes Sanchez Medina Nico Bykhovsky-Gonzalez

Project Delays

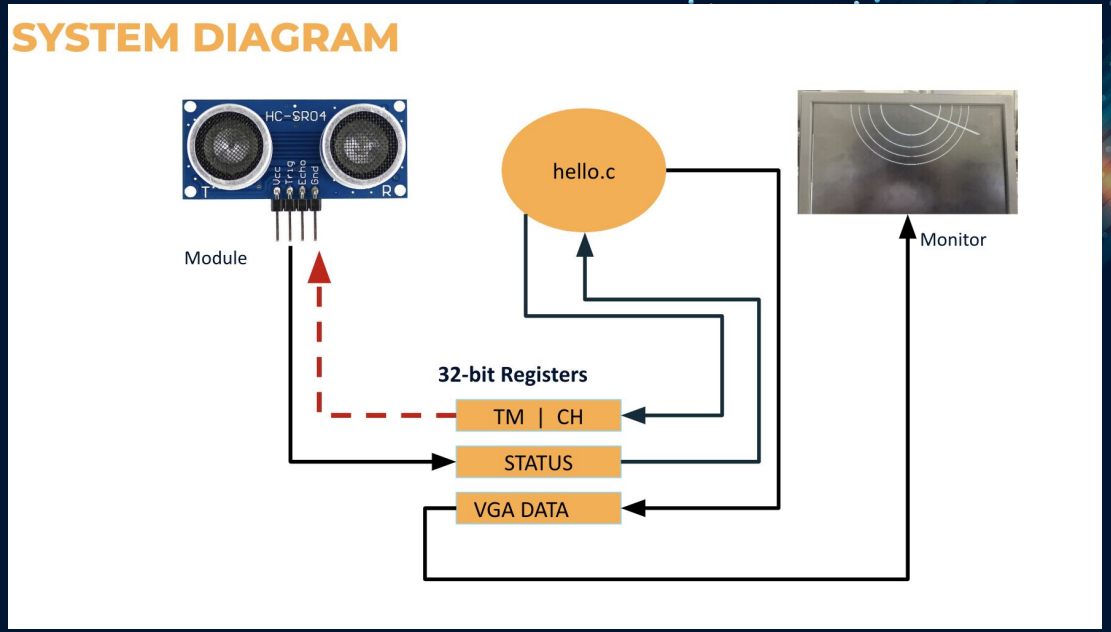
What Works

- Driver between ultrasonic sensor and peripheral
- Adjustable VGA radar according to echo signal (simulated)

Project Delays

What Does Not Work

- Reading an echo signal from UltraSonic Sensor
- Controlling a servo motor with PWM



OVERVIEW

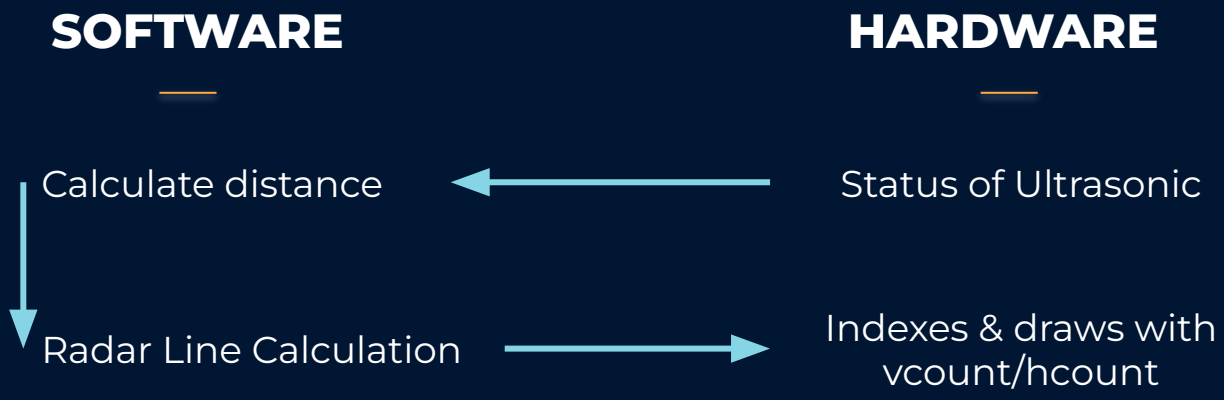
SOFTWARE

Calculate distance

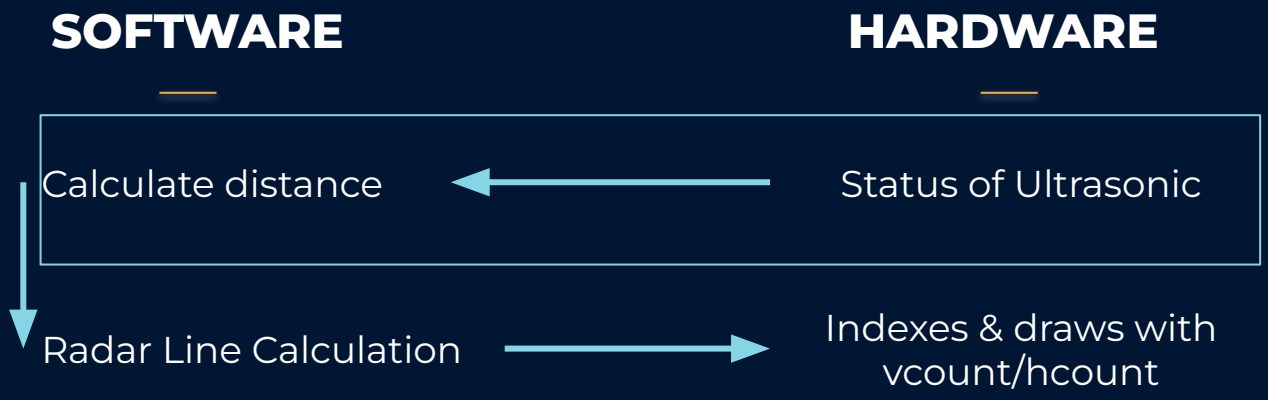
HARDWARE

Status of Ultrasonic

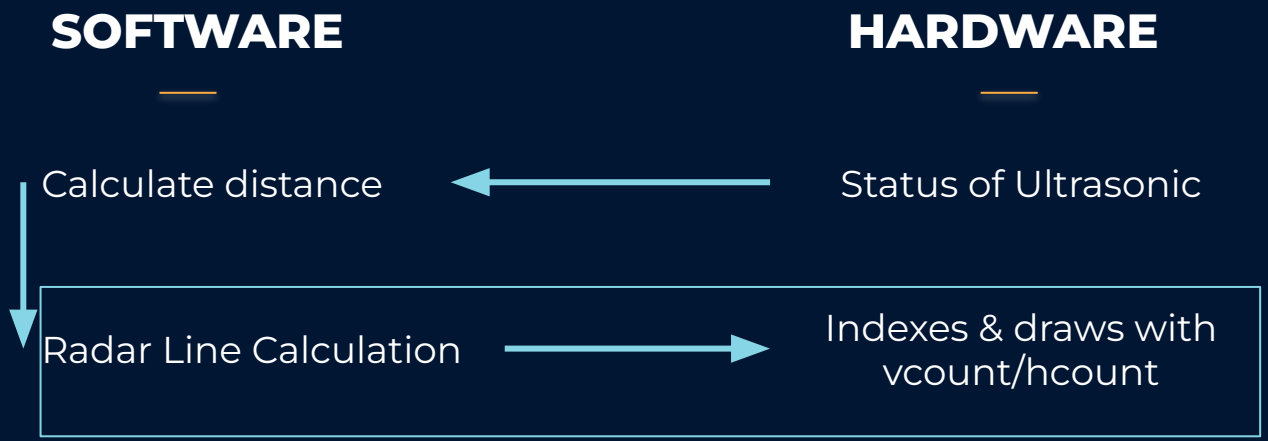
OVERVIEW


SOFTWARE
HARDWARE
Calculate distance
Status of Ultrasonic
Radar Line Calculation
Indexes & draws with vcount/hcount

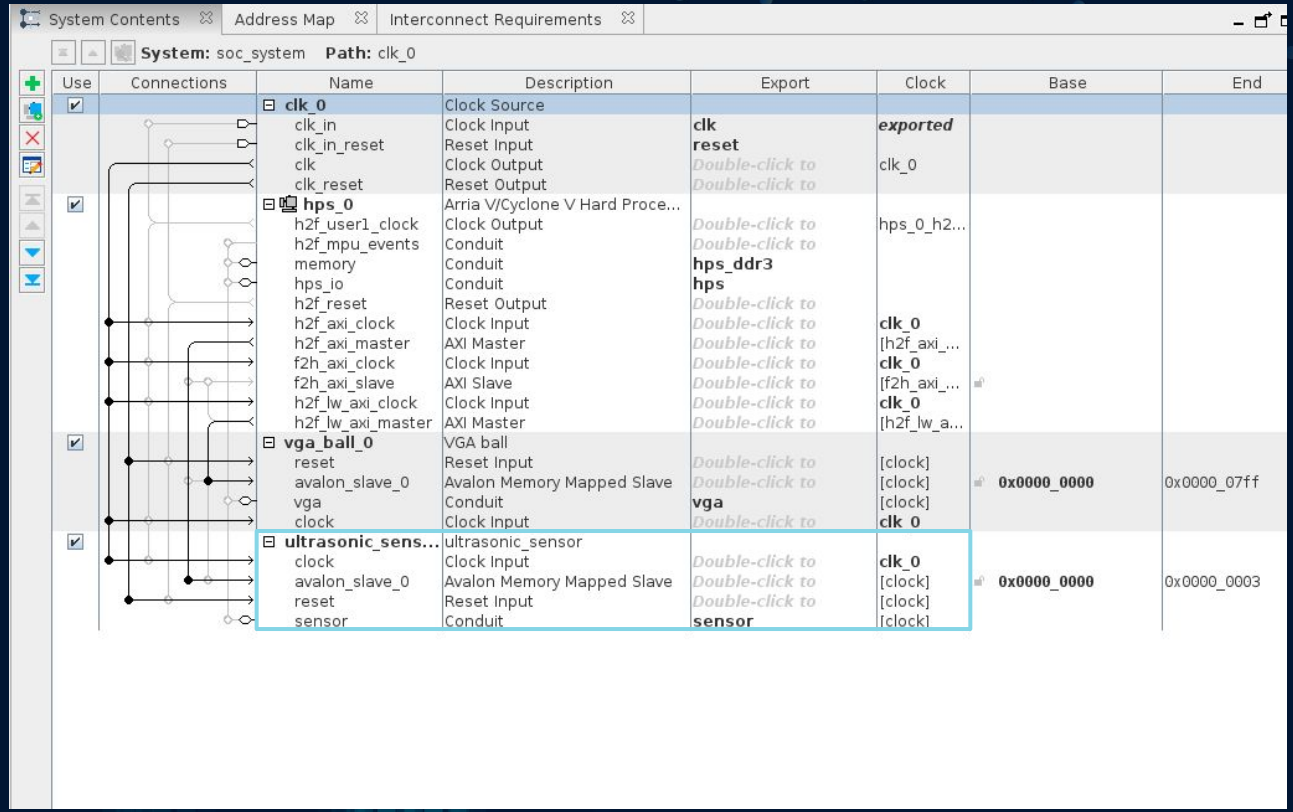
OVERVIEW


SOFTWARE
HARDWARE
Calculate distance
Status of Ultrasonic
Radar Line Calculation
Indexes & draws with vcount/hcount

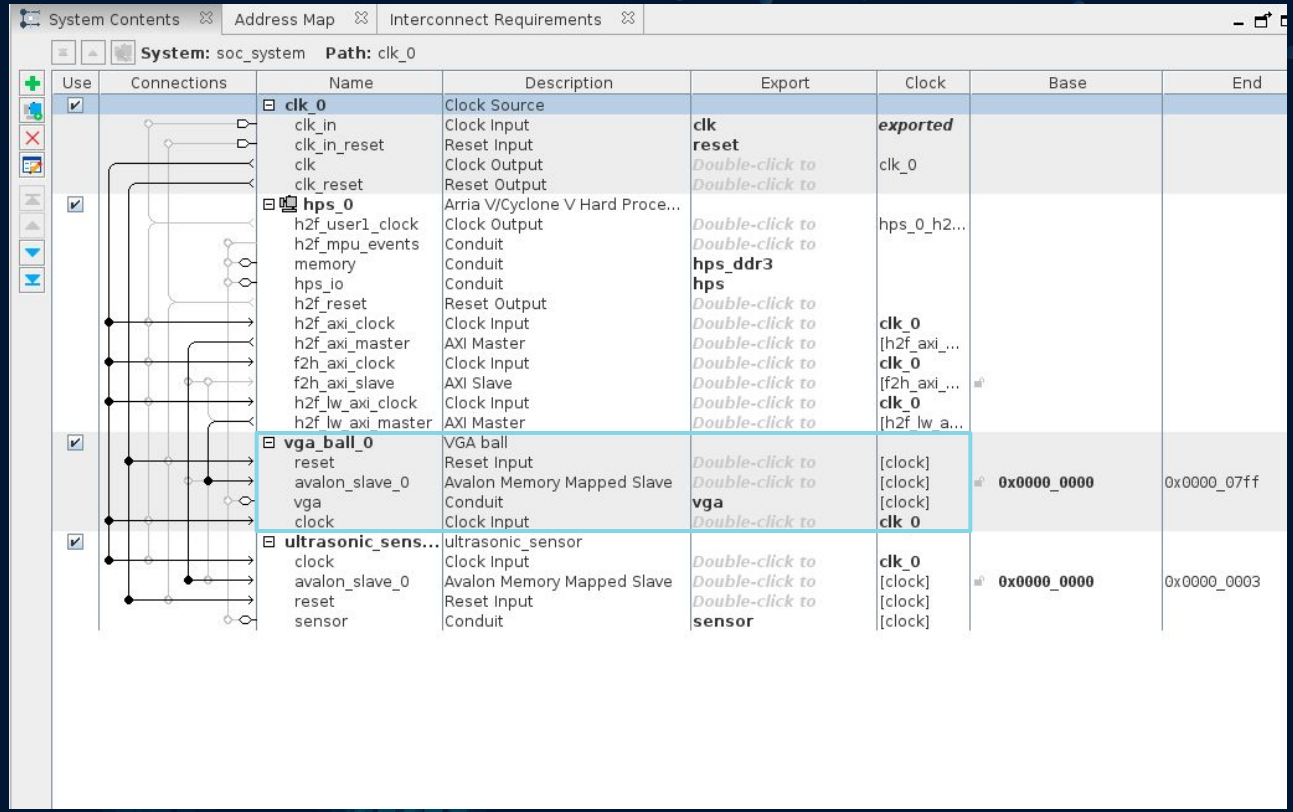
OVERVIEW


SOFTWARE
HARDWARE
Calculate distance
Status of Ultrasonic
Radar Line Calculation
Indexes & draws with vcount/hcount

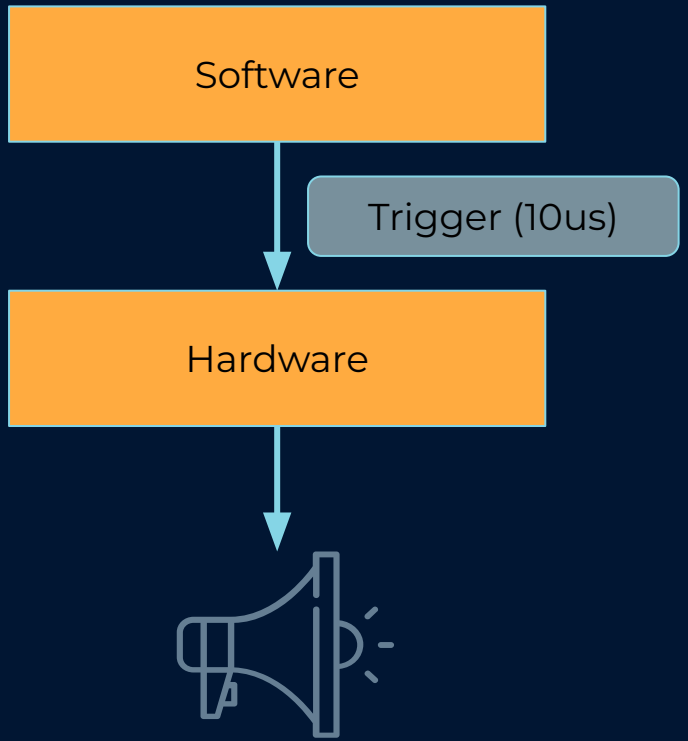
Platform Designer



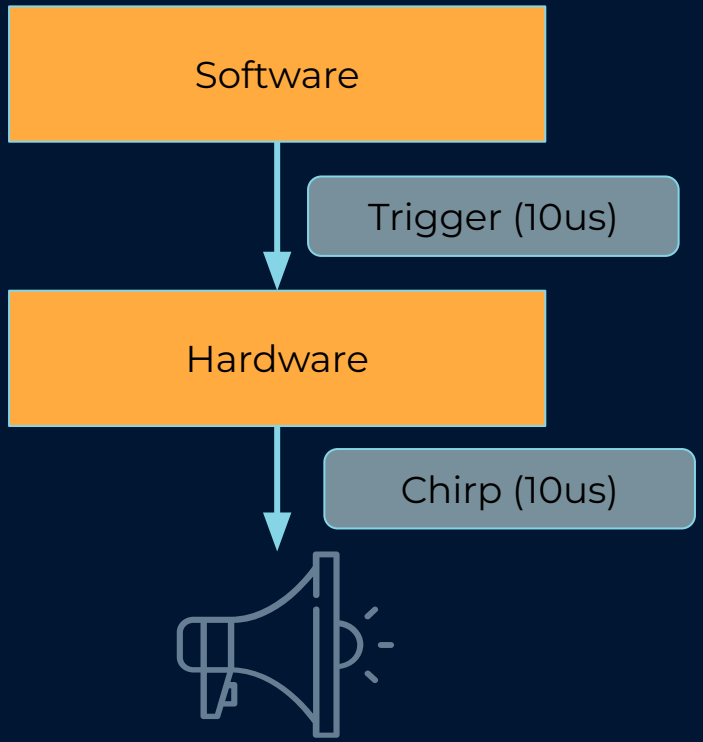
Platform Designer

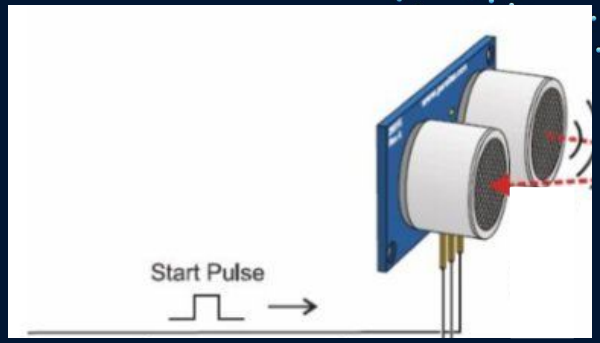


Ultrasonic Sensor Signals

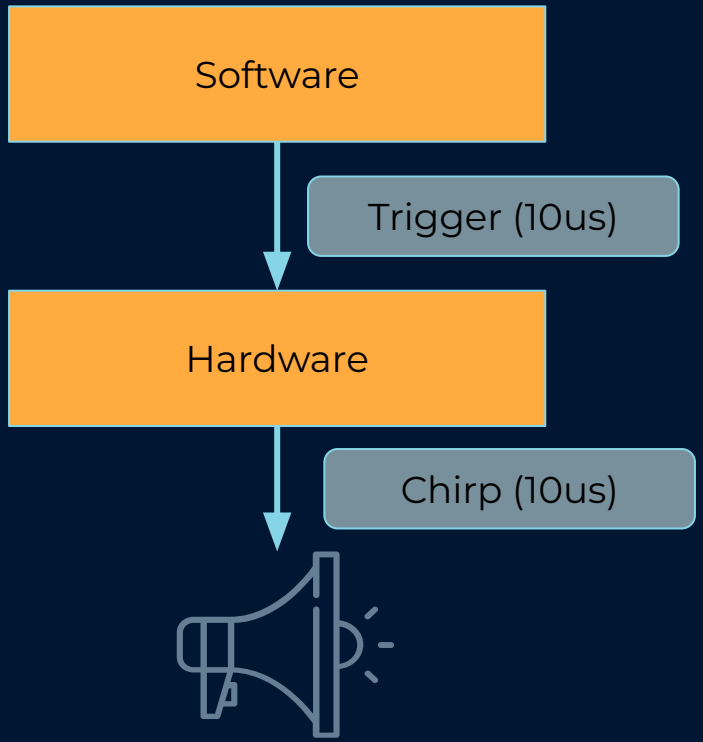


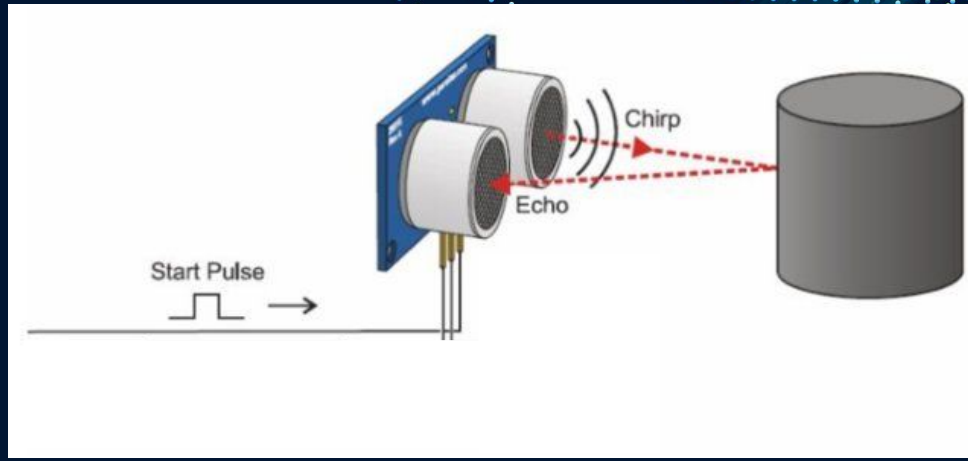
Ultrasonic Sensor Signals



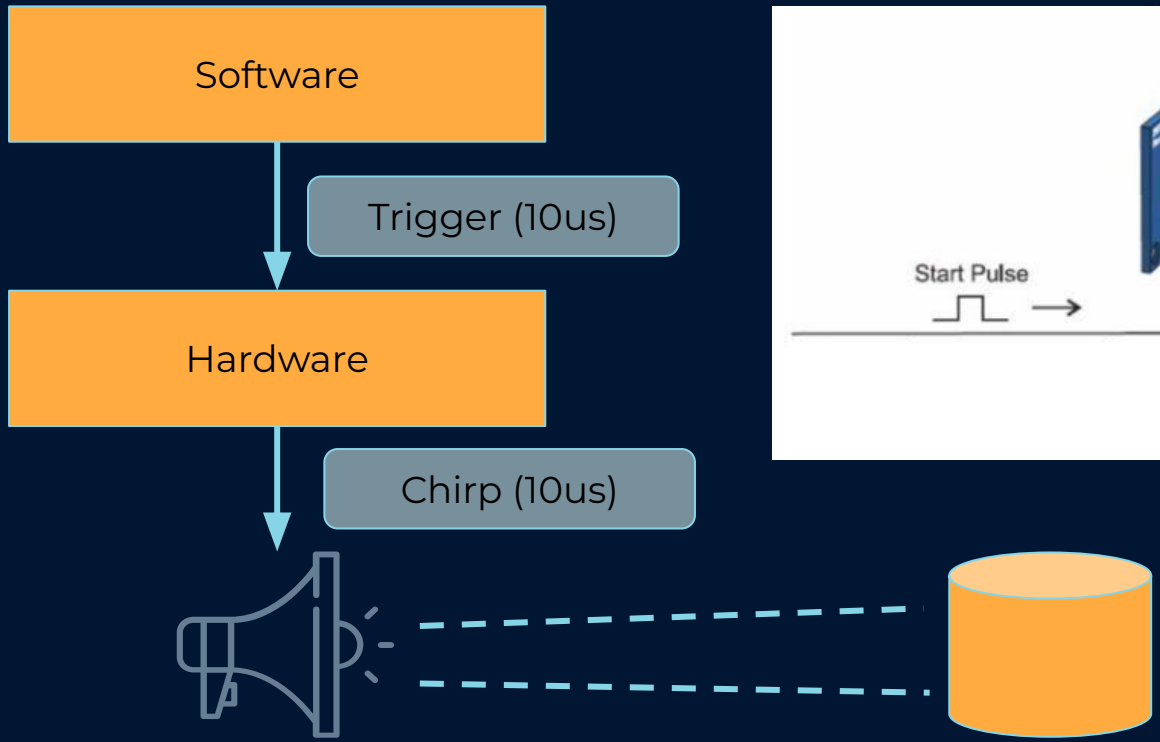


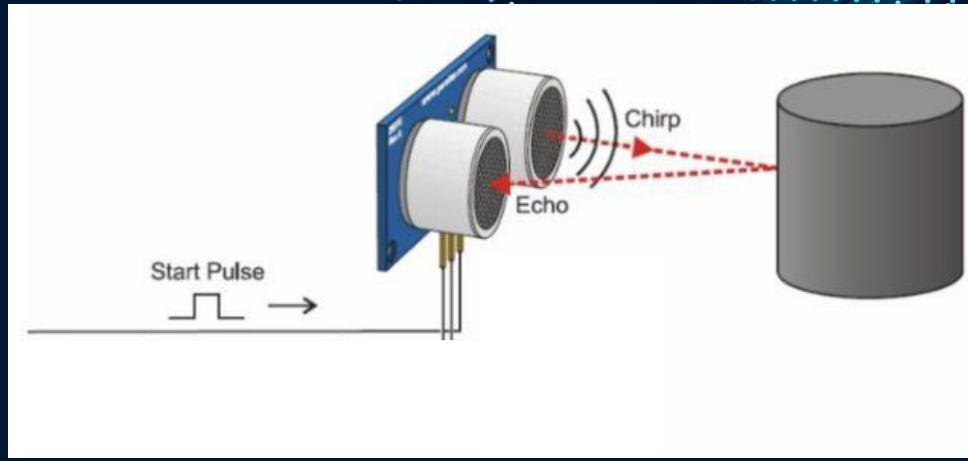
Ultrasonic Sensor Signals



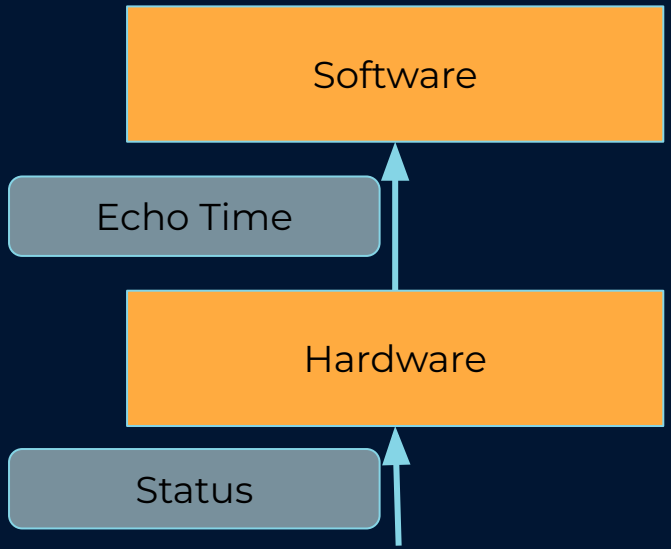


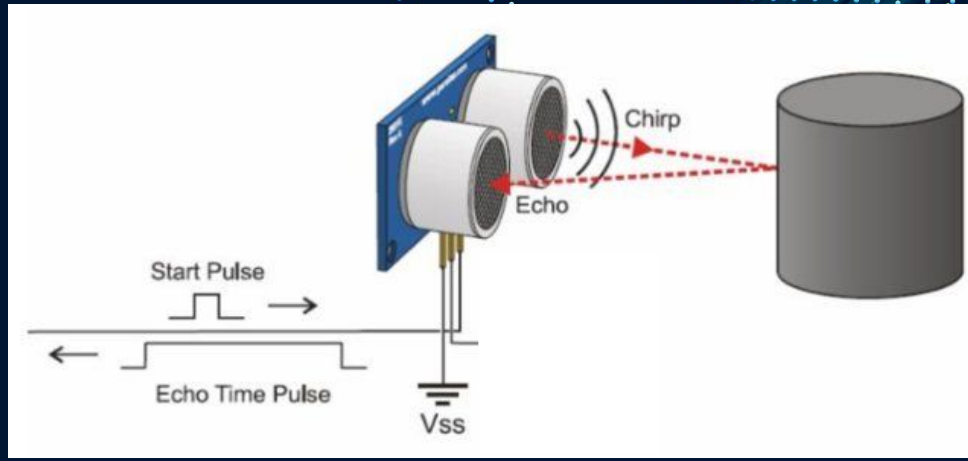
Ultrasonic Sensor Signals

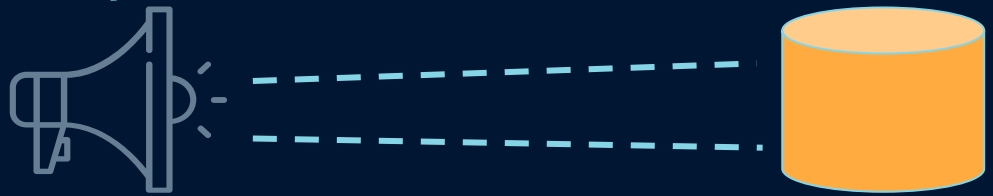




Ultrasonic Sensor Signals

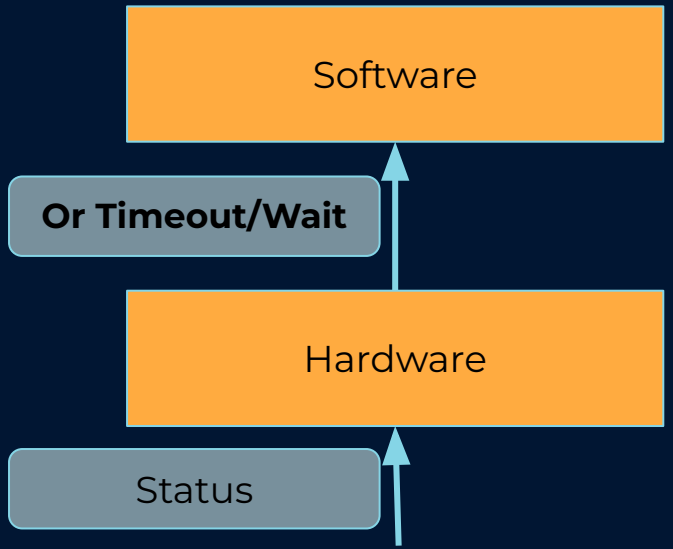


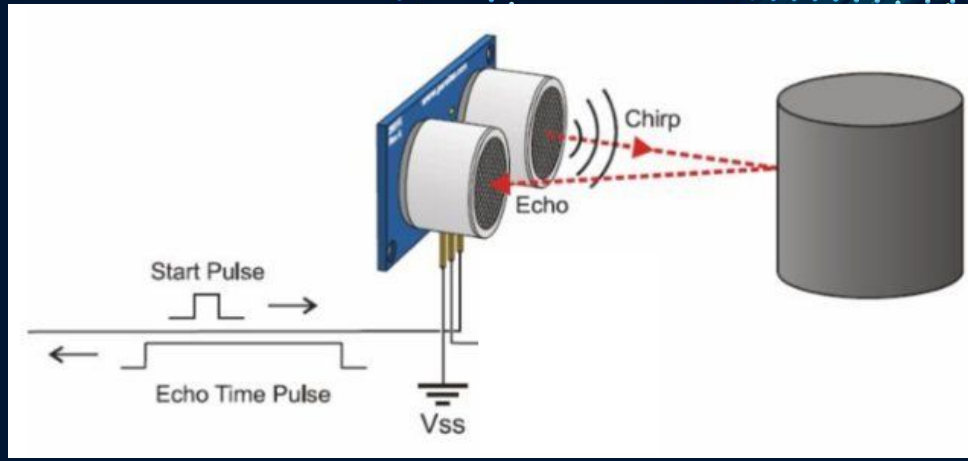


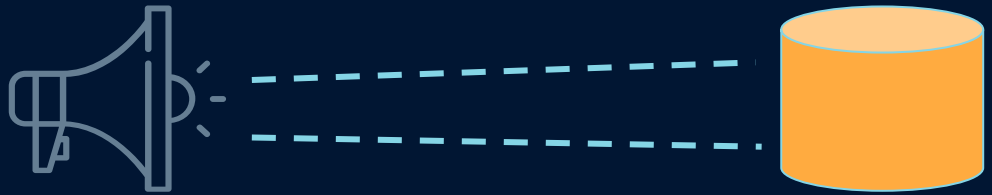


temperature_celsius

Ultrasonic Sensor Signals







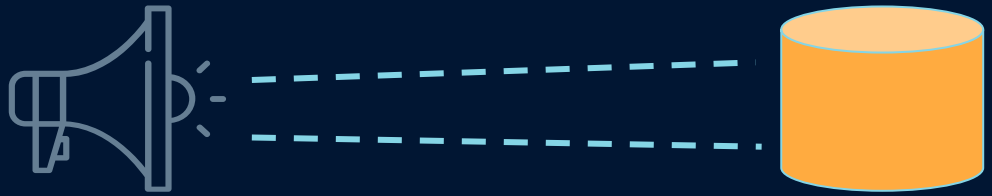
Ultrasonic Sensor Signals

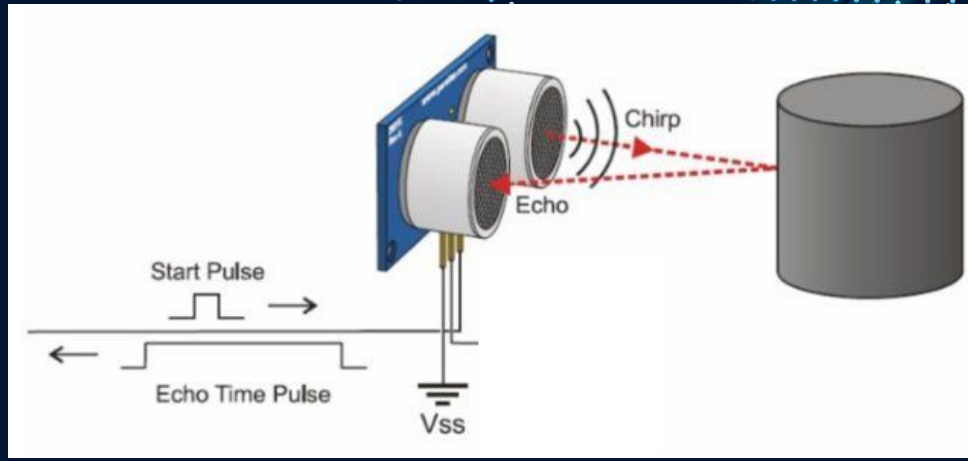
Distance = (SoS × Time) ÷ 2

Or Timeout/Wait

Hardware

Status





Defining the line

Given theta, define a line with horizontal ranges indexable by vcount.

Calculate horizontal distance:

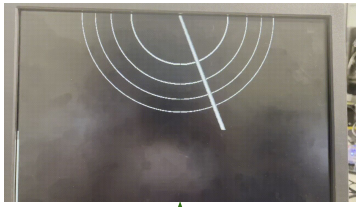
$X_{high/low} = \cos(\Theta) * (\text{skipped x distance}) +/- \Delta$

```
int x0 = SCREEN_WIDTH/2 + (int)(cosf(theta * (float)M_PI / 180.0f) * vx) - (2 + AngleDistanceFrom90);
int x1 = SCREEN_WIDTH/2 + (int)(cosf(theta * (float)M_PI / 180.0f) * vx) + (2 + AngleDistanceFrom90);
```

SYSTEM DIAGRAM

(Module)

hello.c



(VGA Peripheral)

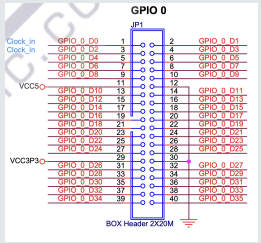
32-bit Registers

TM | CH

STATUS

VGA DATA

ULTRASONIC SENSOR + MODULE



If CHIRP:

→ TRIGGER 10 microseconds

→ Wait for ECHO; STATUS = 0.

If ECHO:

→ Count echo; STATUS = 0.

→ STATUS = Final Count.

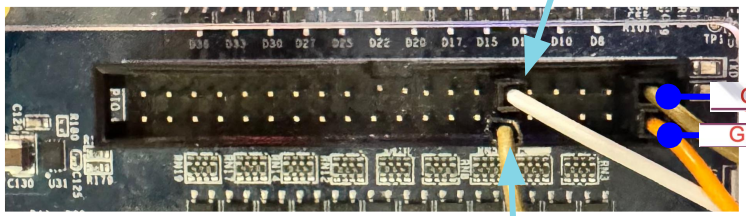
If echo takes too long:

→ STATUS = Timeout.

TIMEOUT | CHIRP

STATUS

ULTRASONIC SENSOR + MODULE

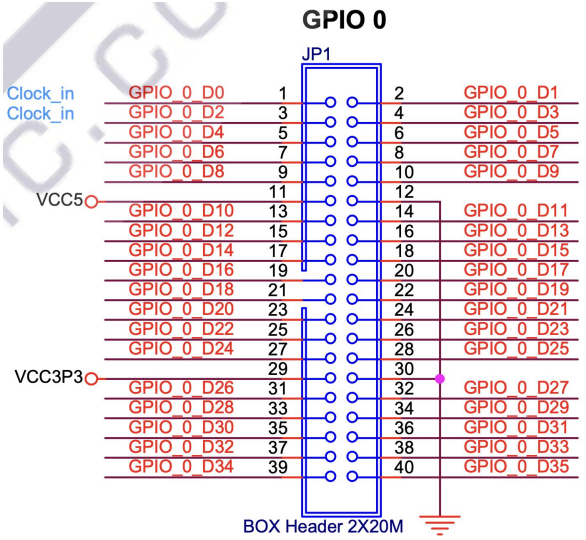


power VCC 5V

GPIO_0_D0

GPIO_0_D2

ground



```
Writing config: timeout=0xffff, chirp=0, cfg=0xffff0000
Echo status @ 132° = 0xe0688c92, chirp = 0
Writing config: timeout=0xffff, chirp=0, cfg=0xffff0000
Echo status @ 131° = 0xe0688c92, chirp = 0
```

```
Writing config: timeout=0xffff, chirp=0, cfg=0xffff0000
Echo status @ 120° = 0xe0abcc74, chirp = 0
Writing config: timeout=0xffff, chirp=0, cfg=0xffff0000
Echo status @ 119° = 0xe0abcc74, chirp = 0
```