

# Demo 1 Design Brief

## I. Identify needs and establish requirements

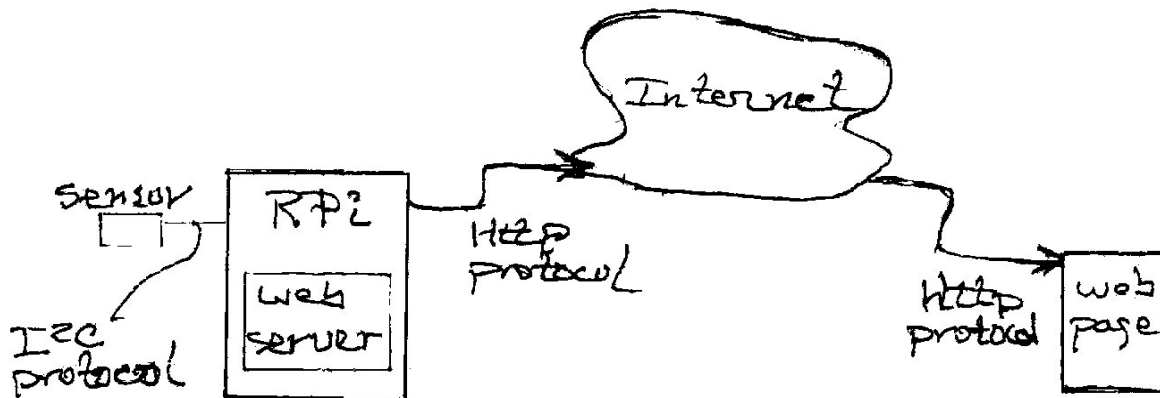
- **Design Objective (i.e., the need):**

Demonstrate the design process by creating a simple “thing” for the Internet of Things (IoT). This demonstration will guide our technical studies and must incorporate all components essential to our study of IoT things. These components are:

- microcontroller
- sensor and/or actuator
- wireless communication
- web interface

In addition to being complete, the demonstration should also be relatively simple and affordable. As a final note, we will **not** consider cloud-based solutions.

- **System Overview**



- **Design Requirements**

To the greatest extent possible the design should meet all of the following criteria:

- Inexpensive
- Mobile
- Small
- Low power
- Low bandwidth
- High speed
- Reliable
- Extensible
- Easy to construct
- Easy to maintain
- Long service life

## II. Develop alternative designs that meet those requirements

There are many possible implementations of the architecture described above. Lists of some of the most promising choices for each component are provided below. In each case, the selected implementation is highlighted. These lists are by no means complete.

It is important to note that the choices made in each category are not independent; for example, the BeagleBone Black could not be paired with WebIOPi nor could the Arduino easily support Flask.

Finally, several design criteria are dependent on implementational details that are not addressed in the choices presented below. Examples include breadboarding vs. soldering, the use of shields and the particulars of of sensor construction.

<b>Microcontroller</b>	<b>Pros</b>	<b>Cons</b>
BeagleBone Black		
Raspberry Pi (RPi)		
Arduino		
PCDuino		
PIC processor		

<b>Device (sensor or actuator) interface</b>	<b>Pros</b>	<b>Cons</b>
I2C		
Analog		
1-wire		
SPI		
Serial		

<b>Wireless communication</b>	<b>Pros</b>	<b>Cons</b>
Bluetooth		
WiFi		
XBee		

Web server	Pros	Cons
Apache		
Lighttpc		
Nginx		

Web framework	Pros	Cons
Flask		
WebIOPi		
Django		
MQTT		

### III. Build interactive versions (so that they can be communicated and assessed)

The RPi-WiFi-I2C-WebIOPi system, and minor variants, will be the only systems constructed.

### IV. Evaluate the designs (measure their acceptability)

Evaluation results are fed back into further design (FEEDBACK / ITERATIVE DESIGN PROCESS)

Criteria	RPi-WiFi-I2C-WebIOPi	RPi-Bluetooth-I2C-WebIOPi	RPi-WiFi-I2C-adhoc
Inexpensive			
Mobile			
Small			
Low power			
Low bandwidth			
High speed			
Reliable			
Extensible			
Easy to construct			
Easy to maintain			

Long service life			
----------------------	--	--	--

All systems will be evaluated on a scale of 1-4 with respect to each criteria.