KOREA UNIVERSITY

Flicker: Rapid Prototyping for the Batteryless Internet-of-Thing

Josiah Hester, Jacob Sorber

Northwestern Univ, Clemson Univ

Sensor Network | 컴퓨터공학과 / **박휘재**

Introduction: Motivation

- The vision of IoT changes such as how we interact with health, water, buildings and nature
- There is always an energy problem with the IoT
  - Battery volume and life cycle costs replacement and maintenance
- There are a number of technologies to manage the system efficiently, but they still require expertise

Introduction: Problem definition

- Limited hardware options
  - Intel WISP and its subsequent products were not well supported in this area, but were used by this task
- Limited flexibility
  - Hardware with various hardware adjustments
- LackingModern Amenities
  - Troubleshooting RTC failures due to lack of hardware power
- Low usability or community
  - Activating communities available to novice developers

Introduction: Solution

- **Flicker**: Flexible, Modular Hardware Platform to Collect Energy
  - Battery-free sensor unit with intermittent power supply
  - IoT 장치를 만드는 사용자에게 빠른 프로토타이핑을 제공
- Component module
  1. Computing Core
  2. Energy Harvester
  3. Sensor module
  4. Communication peripherals

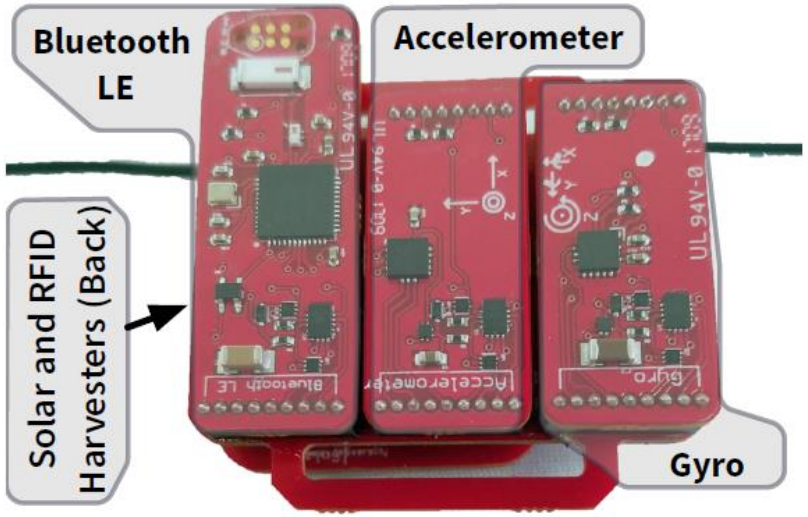
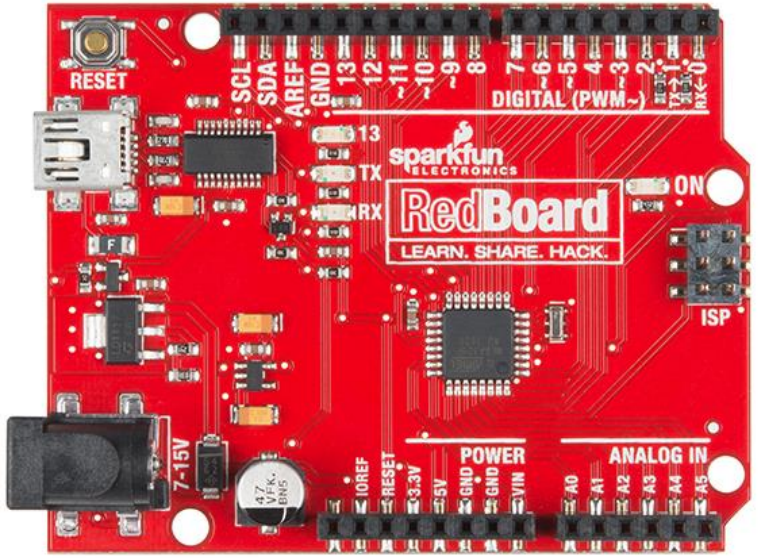


Figure 1: Flicker enables rapid prototyping of batteryless, intermittently powered sensing systems. This figure shows a Flicker device equipped to harvest RFID and solar energy (harvesters are placed on the back side), sense acceleration and rotational velocity, and communicate via Bluetooth LE.

Building: Executive solution

- Traditional solutions for rapid prototyping
  - Sparkfun / Arduino shield
  - Both boards do not have the lowest power consumption and cannot be replaced with solar panels
  - Operating system and language support limited



▲ Red Board



▲ Arduino UNO

Building: Federated Energy

- Allocate dedicated capacitors to all hardware peripherals
- Capacity to store enough energy for a **single job**
- Avoid situations where the sensor is completely inoperable

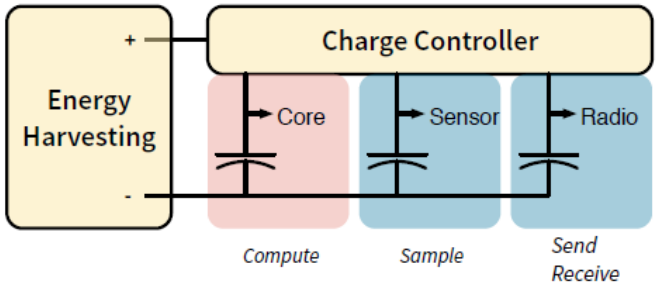


Figure 2: A conceptual view of federated energy storage for capacitor based, energy harvesting sensors. Energy storage is separated per peripheral. Each peripheral maps to a set of of sensing tasks, making energy management more straightforward.

Building: Federated Energy

- Faster device charging
  - Collect not the energy required to support all components but only the energy required for each component
- Tune to Peripheral Specification
  - Allow peripheral devices with different operating voltages to operate on the same device
- Traditional methods take a long time to store the energy required to execute

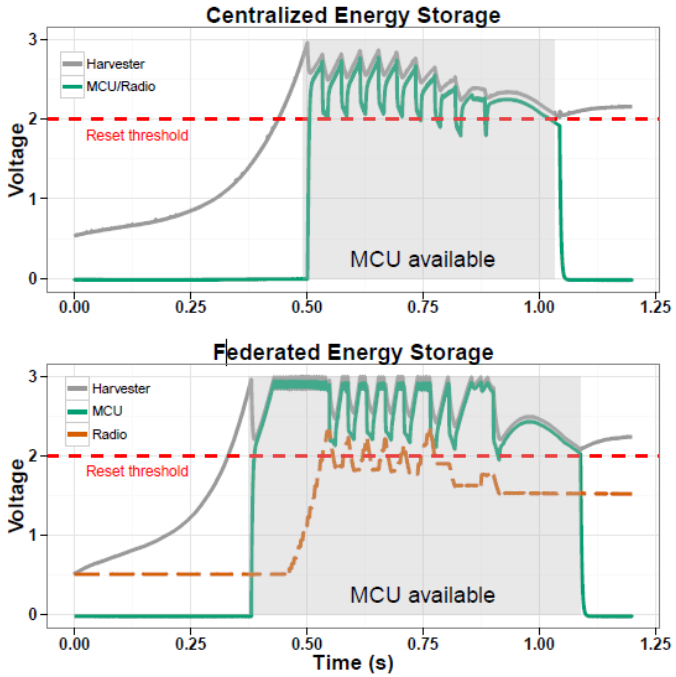


Figure 3: This figure shows a "sense-and-send" program executing with federated energy vs centralized energy storage. Federated energy storage (bottom) allows useful work to start sooner, as smaller capacitors sized to a particular peripheral charge faster than a single shared reservoir. FedEnergy also reduces power failures from coupling of hardware peripherals, and harvests more energy.

Building: Effect

1. Increase flexibility and efficiency of integrated energy storage devices
2. Incorporate battery-free IoT into new platforms for flexible and rapid prototyping

Flicker: Summery

- Target
  - IoT App Designer to Develop Energy Harvesters Without Battery
- Goal
  - Provides a range of hardware options in terms of peripherals and acquisition technology
  - Achieve flexibility in runtime and design time
  - Enables timekeeping and energy management on a full-purpose platform
  - Provides a platform focused on overall hardware stack usefulness

Flicker: Summery

- Structure of Flicker
  - Computing core module
    - Port for connecting MCU and peripherals
  - Peripheral
    - Connect digitally to interface ports
    - Digital bus lines such as SPI, I2C, and UART
  - Energy harvestor
    - Collect and send various energy to each capacitor

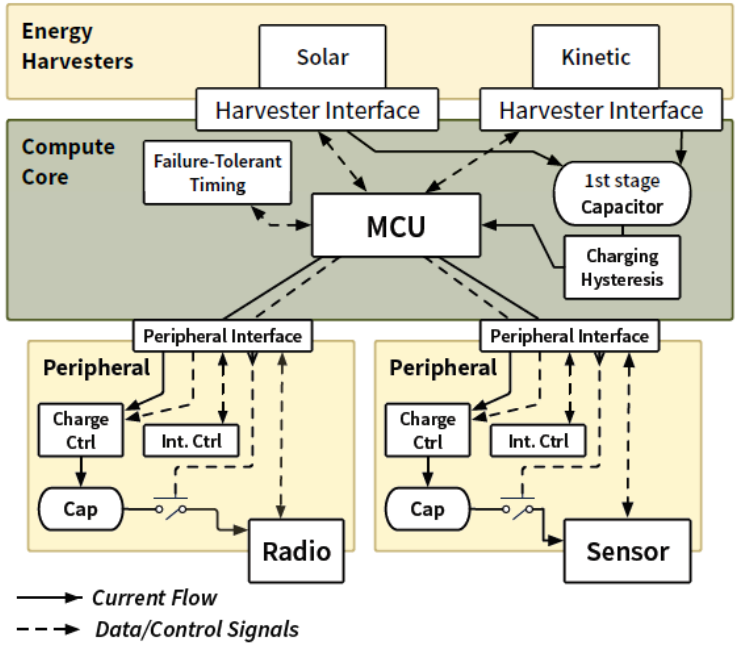


Figure 4: Flicker harware architecture. A multi source energy harvesting interface feeds into the first stage capacitor that powers the Compute Board. Peripherals are connected through a standard interface that maps power, and control signals.

Flicker: Detail research

- Reconfigurable **Federated Energy**
  - 프로그래머가 주변 장치에 커패시터에서 수집할 에너지 양을 할당 가능
  - Allows allocation of the charge points(Trip point)
  - Develop mechanisms to keep overhead low
    1. Transfer to energy polling "Peripheral Ready" interrupt signal
    2. Changing the voltage threshold stored on the capacitor
    3. Use voltage regulation for peripheral devices
  - Capacitors mounted on each peripheral are charged when certain threshold values are reached

Flicker: Detail research

- Peripheral Device Port
  - GPIO, SPI, I2C 와 같은 다양한 연결 옵션 제공

- Critical "Fault-tolerance timing" in intermittent power supply situations
  - Mechanisms that can withstand power failure
    - Remanence-based SRAM timing
    - 맞춤형 회로는 커패시터를 이용해 작동 중단 시간 줄일 수 있음
    - RTC를 이용해 많은 타이밍 정보 제공함

- Circuit that can automatically detect connected modules

Flicker: Workflow

1. Select the configuration you want to use with peripheral devices and the Harvesters
   - 임계 값 전압 및 충전 컨트롤러 및 인터럽트 포함
2. Attach peripherals and collectors to assemble
3. Detects if a wire is connected to the peripheral
4. Create configuration file
5. Repeat the Debugging & Testing
6. Create Own Hardware(Prototyping)

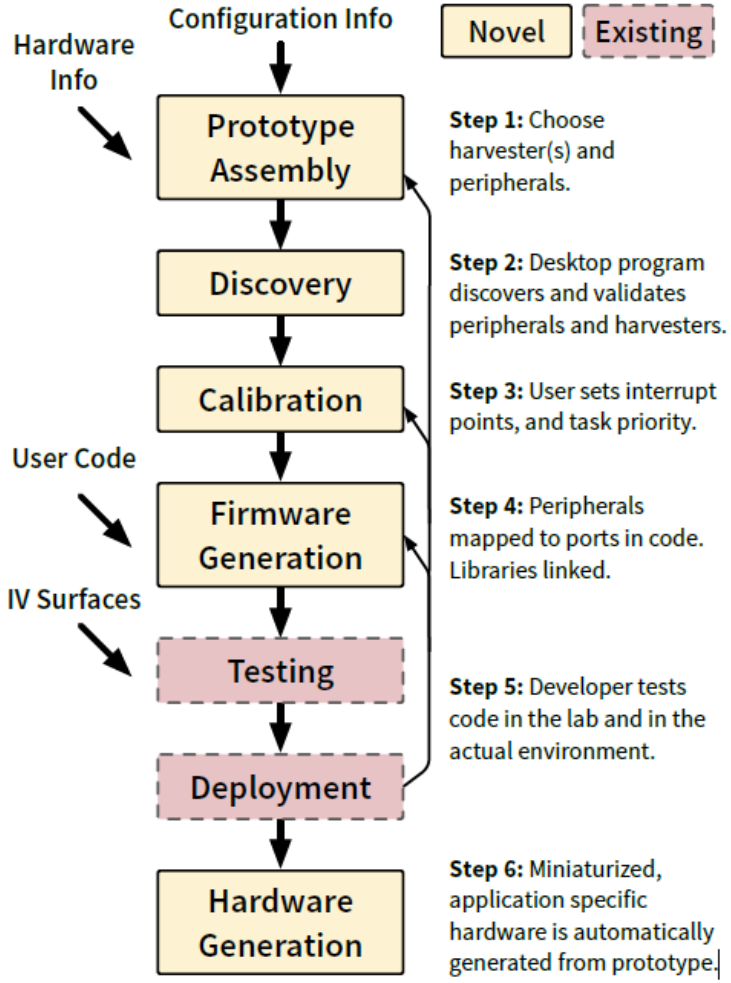


Figure 5: Flicker Workflow.

Implementation: Device config.

- Core module with connector(하베스터 연결 포트 2개+ 주변 장치 연결 포트 3개)
- Tools to facilitate the design process
- Developer Control Runtime Library

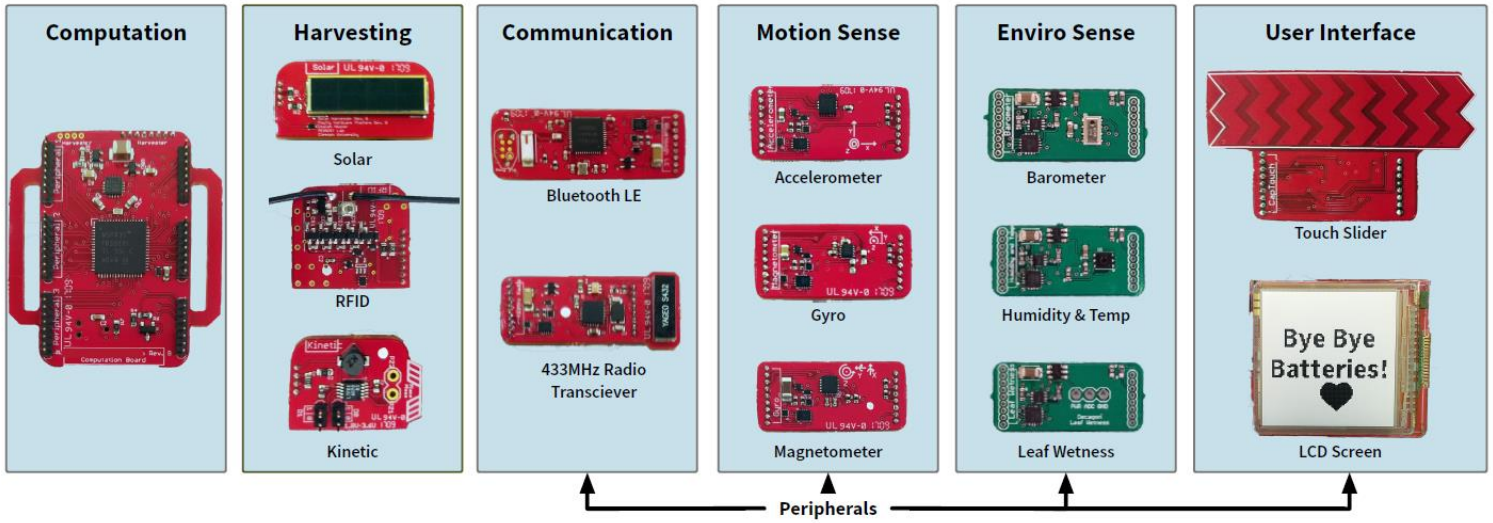


Figure 6: Flicker hardware modules, including a single compute core board, three harvesters, and ten peripherals for sensing, communication, and interacting with users. Peripherals and harvesters attach to the core module, and use federated energy storage with charge and interrupt thresholds to support a wide variety of user tasks.

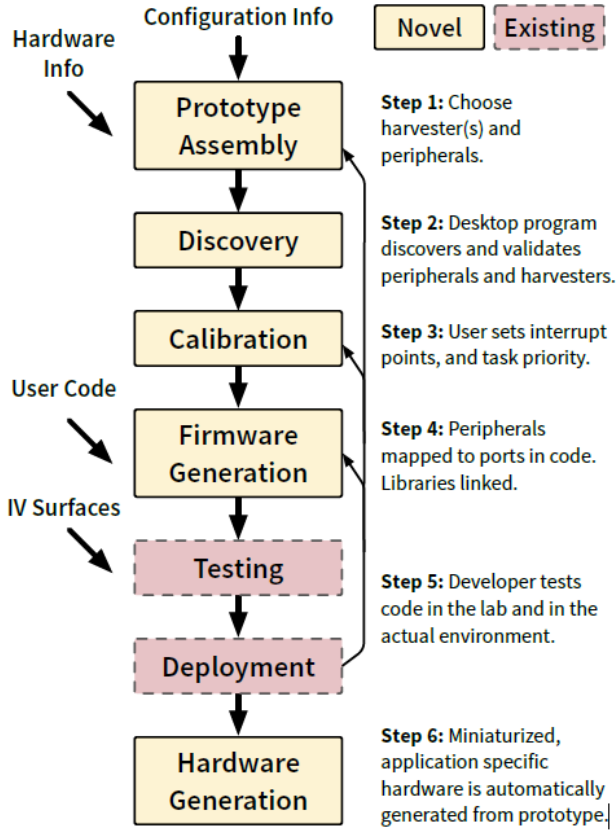
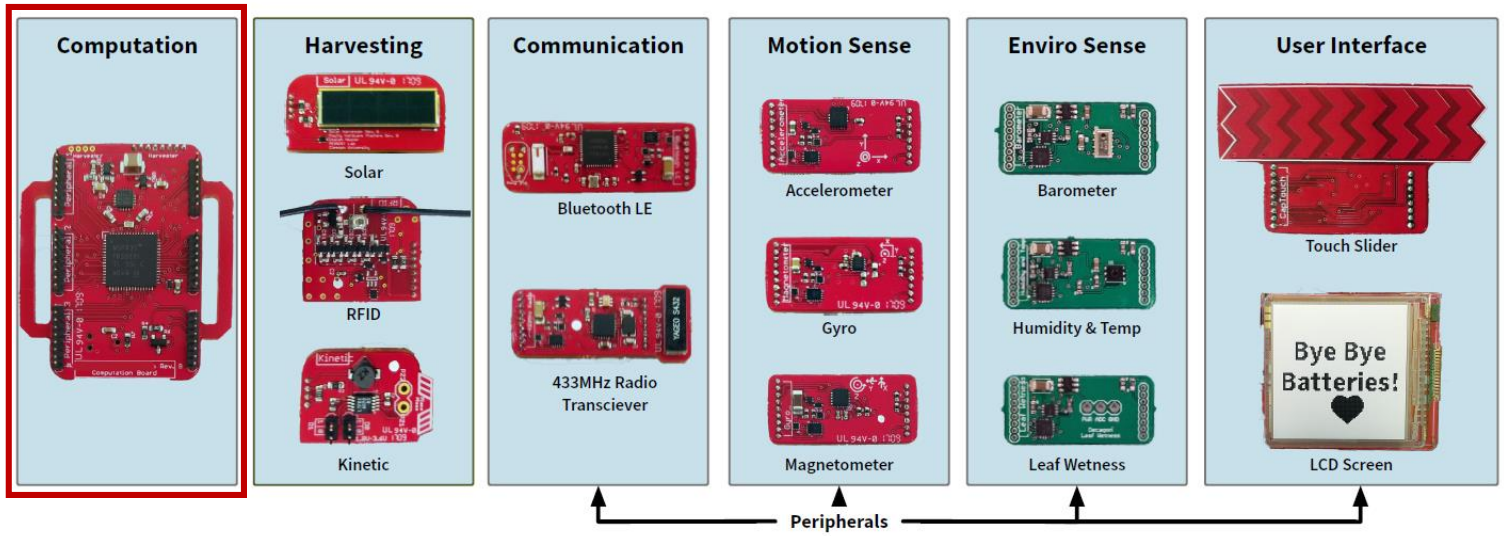


Figure 5: Flicker Workflow.

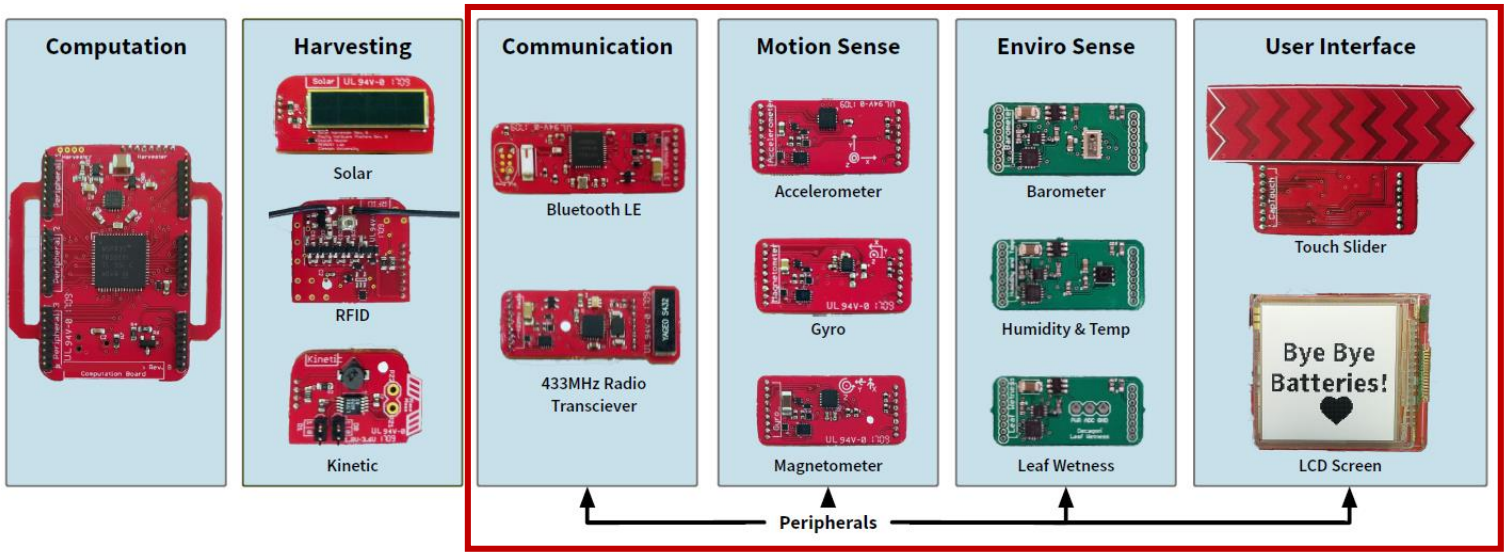
Implementation: Hardware

- Computing Core
  - Multiple ports, but prototypes support one I2C at a time
  - Compile process hangs when connected to incompatible ports
  - Prevent full discharge for up to 19 minutes upon power failure



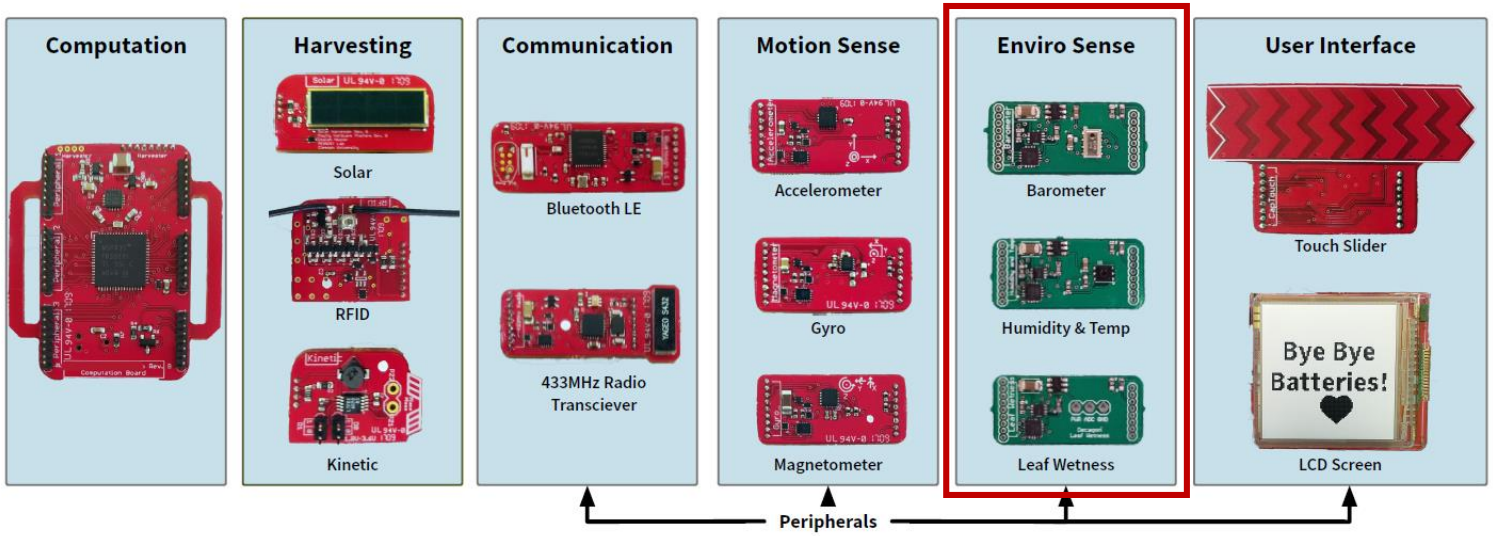
Implementation: Hardware

- Universal peripheral interface
  - Capacitor voltage and peripheral capacitor voltage setting controllers (charge priority control)
  - ✓Interrupt controller, storage capacitor, identification circuit, power gate switch



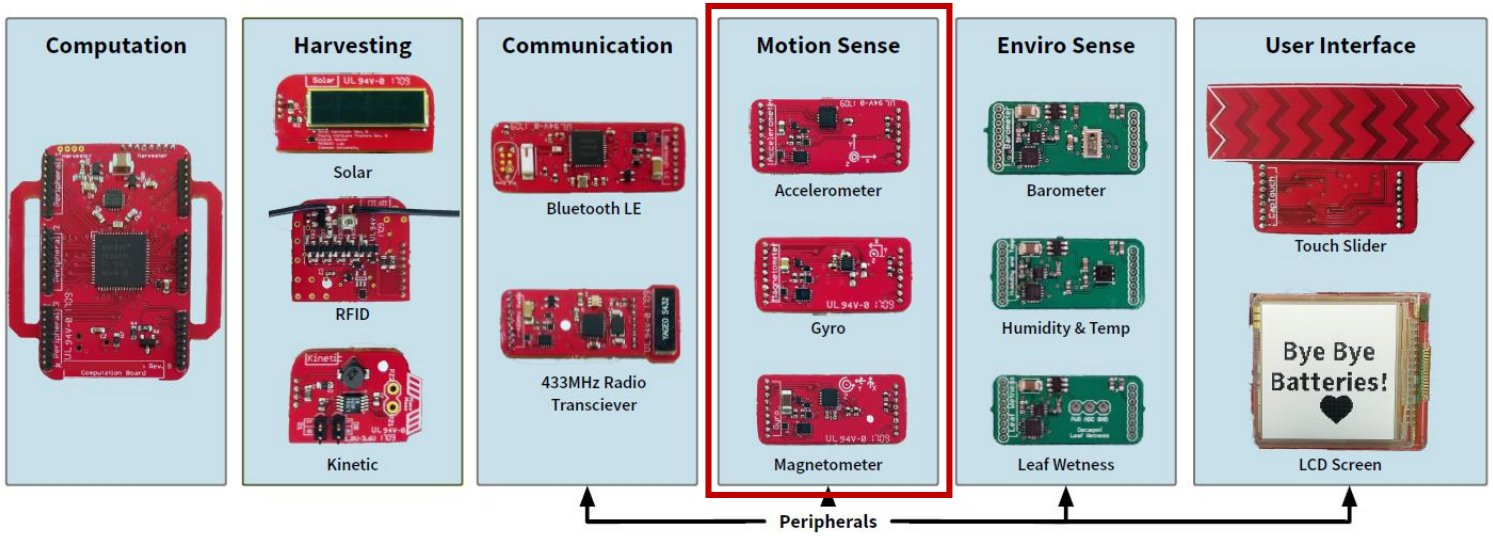
Implementation: Hardware

- Environment sense
  - Collect information about the outdoor environment, such as the detection of greenhouse gases and the placement of topographical space.
  - ✓Composition of low-power barometers, temperature-humidity sensors and plant leaf water sensors



Implementation: Hardware

- Motion sensing
  - For mobile sensor networks, warables, infrastructure monitoring and manufacturing applications
  - ✓Acceleration measurement, gyro for motion detection, geomagnetic detection



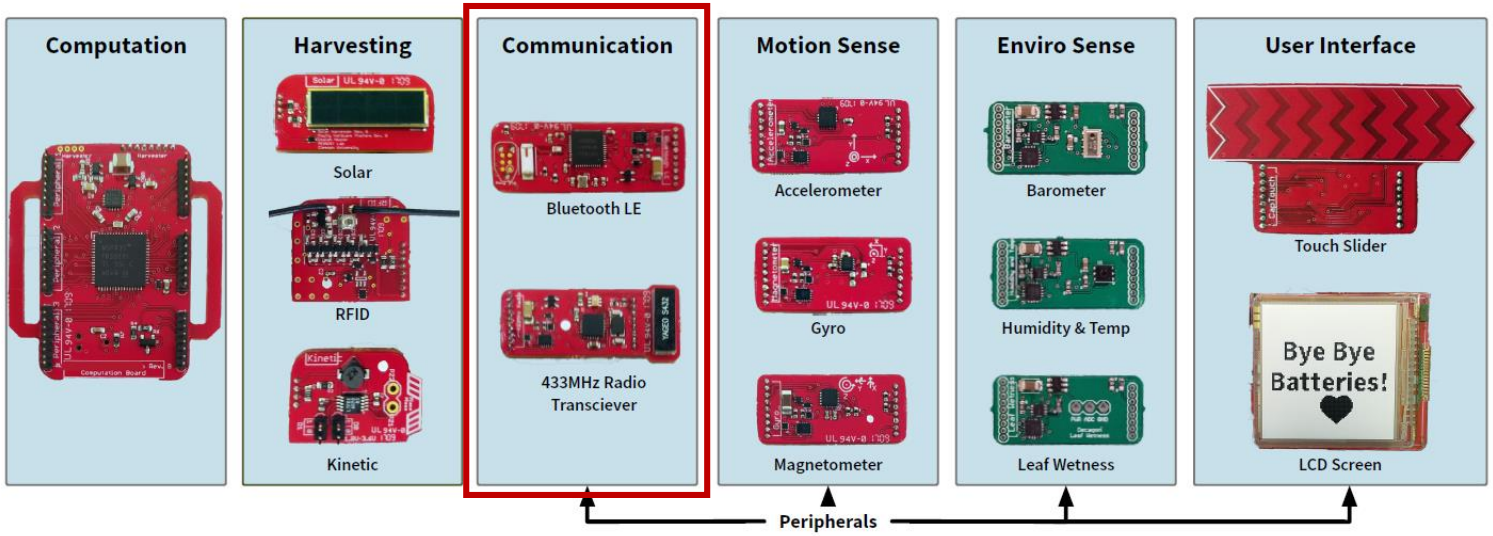
Implementation: Hardware

- Communication
  - IoT의 필수 부분이므로 광범위한 통신 모드 지원해야 함

✓SPI to communicate with MCU using small chip antenna

✓BLE communication between wearables or mobile device

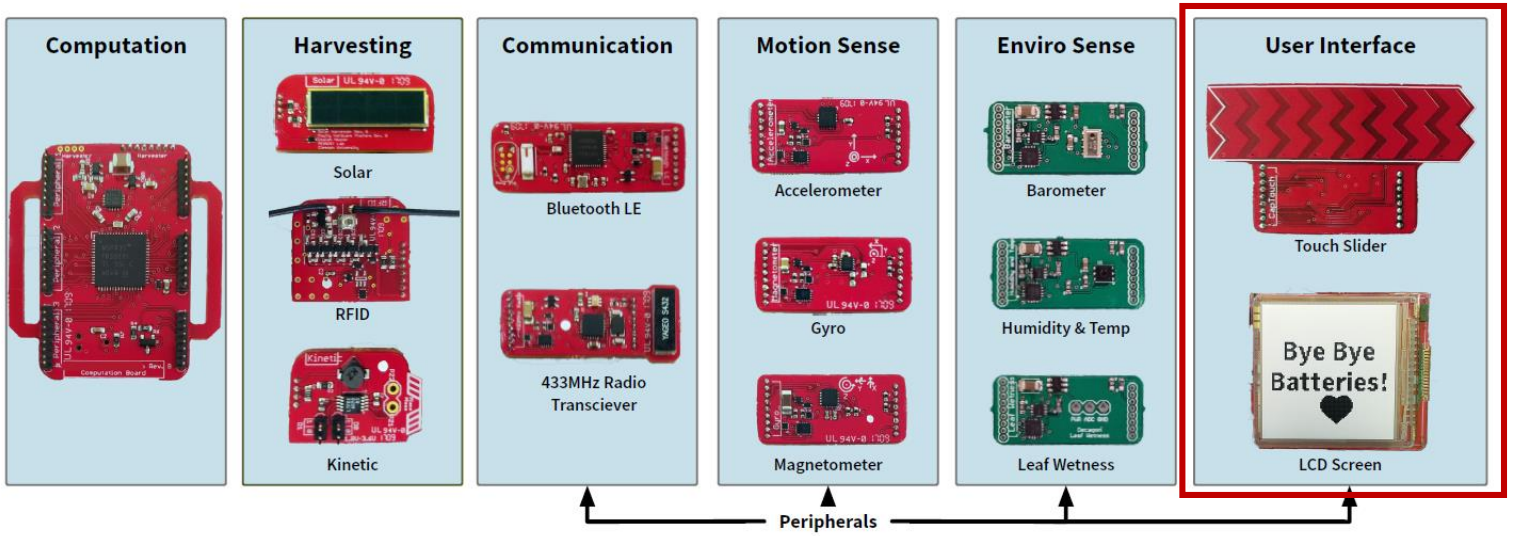
✓Implementation of ultra-low power, two-way water supply communication using RFID



Implementation: Hardware

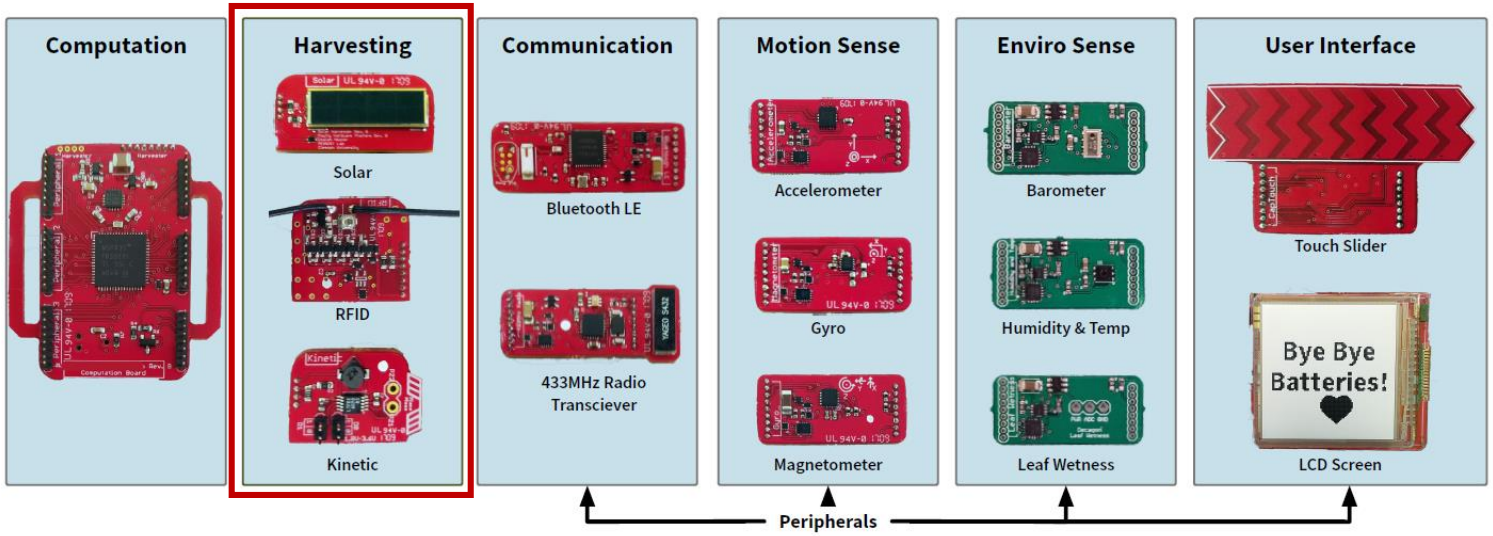
- User interface (UI)
  - UI to check peripherals intermittently

✓Slider with low power display and 8 buttons



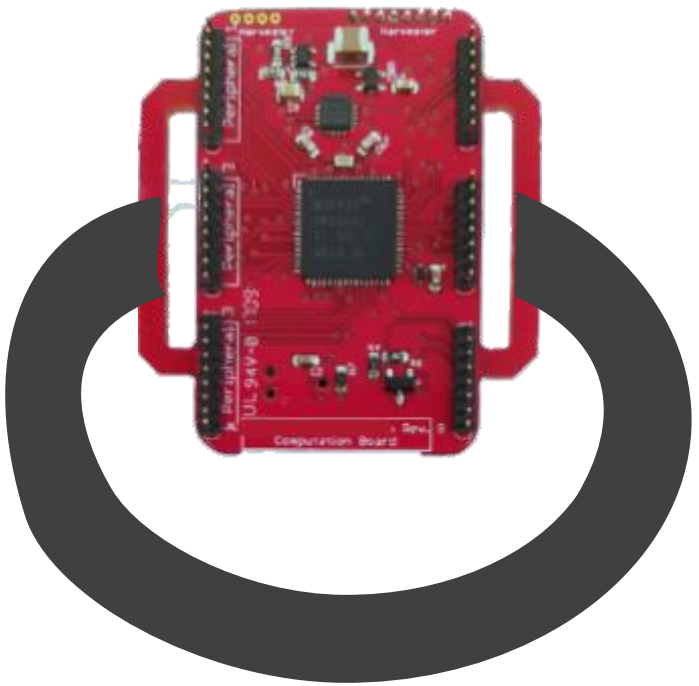
Implementation: Hardware

- Energy Harvestor
  - Three different energy collection schemes and two slots available
  - Solar Collector
  - Low power charger with RFID signal receiver



Implementation: Hardware

- Standardized hardware templates and interfaces for easy user access
- Increase durability using connectors larger than the size of the circuit, and enable clock bands
- 연결한 모듈에 따라 비용은 다르지만 부품 및 인쇄 회로 보드의 비용은 한 세트에 200달러 내외
  - 사용자가 직접 만드는 것 보다 저렴



Implementation: Software

- Firmware support to manage each step of the prototyping pipeline
- Save development time, runtime, and deployment time

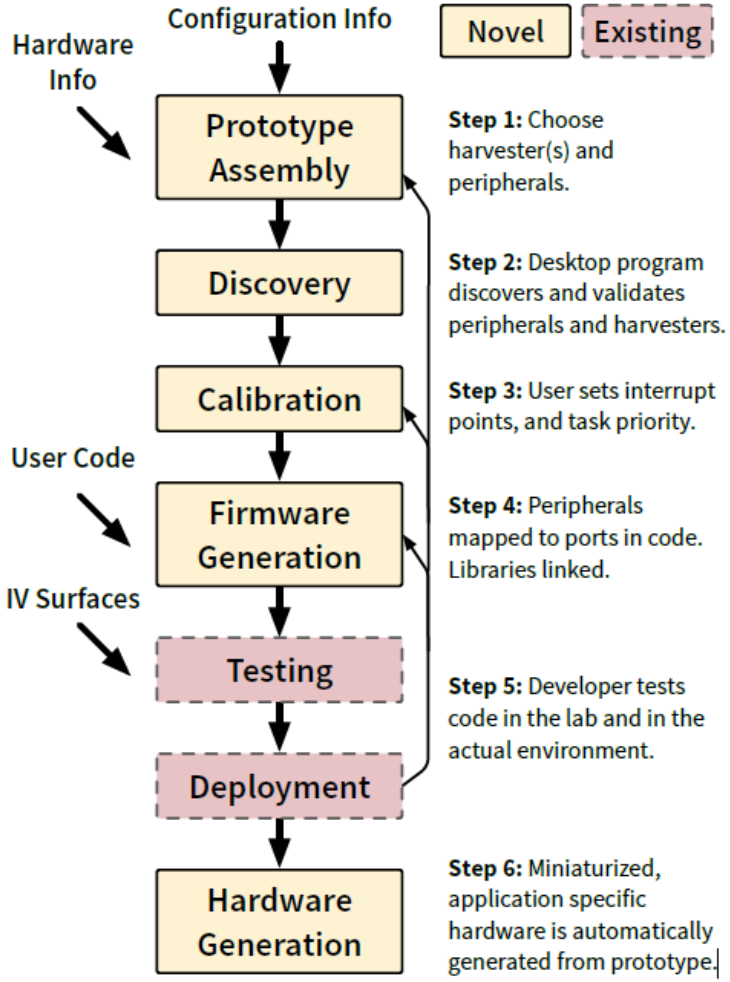


Figure 5: Flicker Workflow.

Implementation: Software

- Discovery and calibration
  - Upload specially discovered firmware when the firmware creation process starts
  - Identify the module fitted
  - Store in a predetermined memory location on MSP430
  - Register recording after error check by tool chain

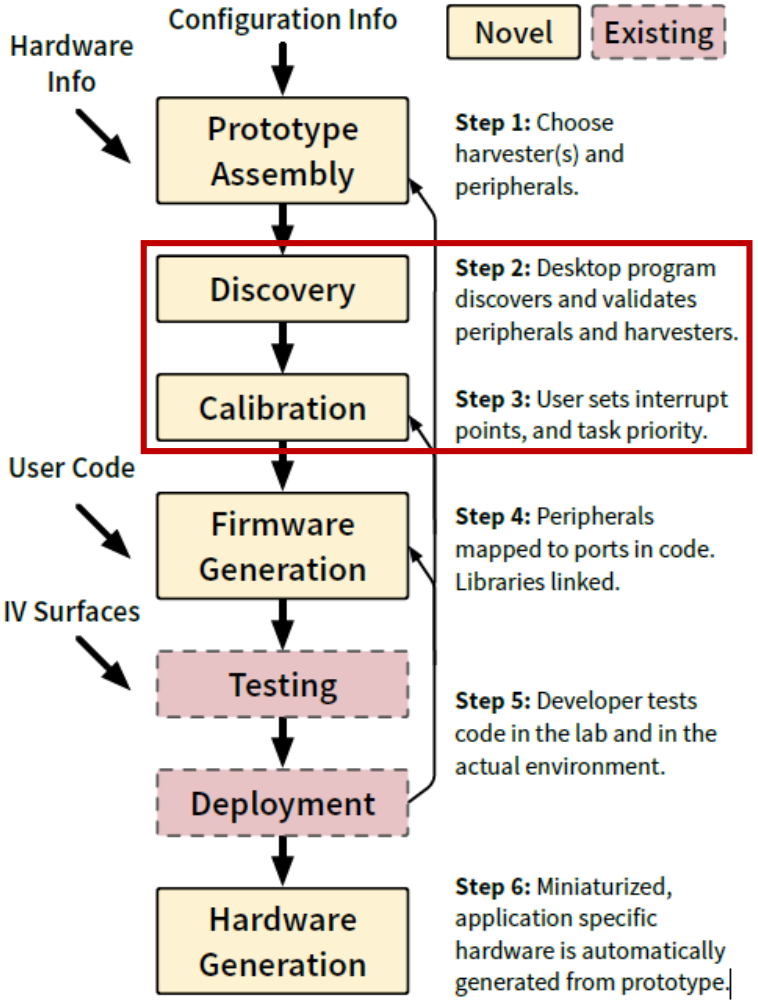


Figure 5: Flicker Workflow.

Implementation: Software

- Runtime library
  - 주변 모듈과 함께 사용 가능한 라이브러리 개발
  - 기본 감지 및 통신 작업 수행 가능
- Ambient energy priority control library
  - Set interrupt level and charge threshold value using simple API
- Automate design
  - 설계 자동화를 이용하여 주변 장치, 수집기, MCU를 결합
  - Python과 EAGLE CAD 스크립트 이용하여 구현
  - 적절한 저항 설정을 위한 회로도 생성

Evaluation

- Evaluating Flicker's Overhead and Performance
- Quantifying the usefulness of federated energy
- Evaluation of flicker overhead in terms of energy and user time
- Assessing usability points of view that prototyping has been simplified

Evaluation: Overhead

- Need to be efficient because it is influenced by the energy gathering
  - 충전시간과 에너지 비용 간의 합의점인 10uF 커패시터를 사용
- Overhead tables exclude some energy efficiency for the Flickers' flexibility
- The table shows that all values are low

Table 1: Flicker overhead

Parameter	Value
Timekeeper Charge Energy	39 µJ
Timekeeper Startup Time	1.1 s
Volatile Threshold Write Time	197.7 µs
Volatile Threshold Write Energy	69.3 nJ
Peripheral Voltage Range	1.7 to 5.5 V
Peripheral Current Range	0.0 to 40 mA
Compute Board Quiescent Current	5.77 µA
Peripheral Quiescent Current	4.47 µA

Evaluation: Performance

- RF Harvester
  - RFID 판독기에 연결된 소형 안테나를 1cm이내에 놓았을 때 최대 전력 1.4mW, 최대 전압 6.3V, 최대 전류 1.0mA
  - RFID 구동 어플리케이션인 Moo 나 WISP 플랫폼과 유사한 전력 레벨을 보임

- Timekeeping
  - 커패시터가 RTC의 완전 방전까지 19분 40초 방지
  - 방전 후 전원 켜는데 1.1초 걸림
  - I/O 이용하여 신호 보낼 때 까지 휴면 상태

Evaluation: User study

- 대학교 컴퓨터 OS 소속 19명의 고학년
- 30분 세션에 참여해 76개의 장치 만듦
- 평균 9.5시간 보냄

- 산업 표준 시스템인 사용 적합성 조사(SUS) 결과에 따라 뛰어난 유용성 평가 결과

Table 2: System Usability Survey (SUS) scores vs. Flicker

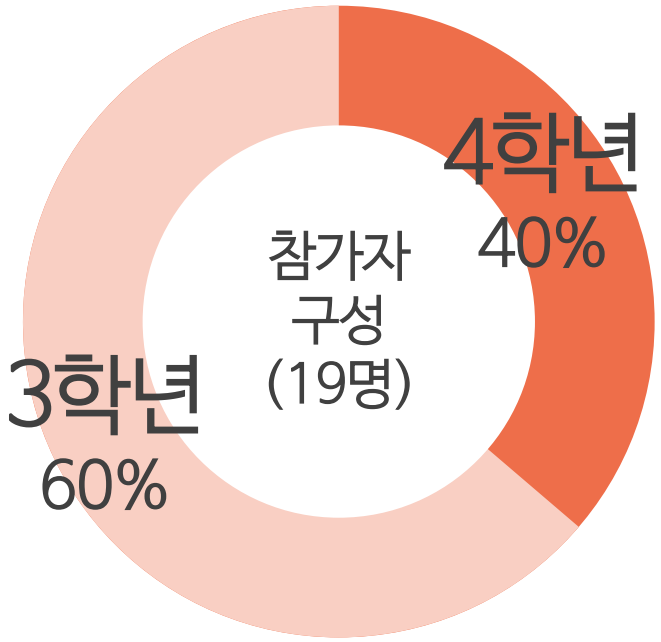
Interface Type	SUS Score
Cell Phone	66.55
Customer Premise Equipment	71.60
Graphical User Interface	75.24
Interactive Voice Response Systems	73.84
Web Pages and Applications	68.05
Flicker Platform	**84.9**

Evaluation: User study

- 제공되는 구성물
  1. 스스로 자신의 역량 평가 후 전자 장치 플랫폼에 대한 경험 기술
  2. IoT에 대한 설명과 IoT 기기 구축, 프로토타입 제작에 대한 동기, 애플리케이션 및 어려움에 대한 간략한 개요를 포함하여 배터리 없는 장치 제작 약속문
  3. 플리커에 대한 안내문과 연산 보드, 그외 앞서 설명된 모듈 일체

- 출구 조사 설문 내용
  1. 소프트웨어/ 하드웨어 / 휴대전화 / 웨어러블 및 웹 사이트 사용 용이성을 측정하고자 시스템 사용 가능성 조사가 포함됨
  2. 사전지식, 플랫폼에 대한 향후 관심, 경험에 대한 즐거움이나 혐오감

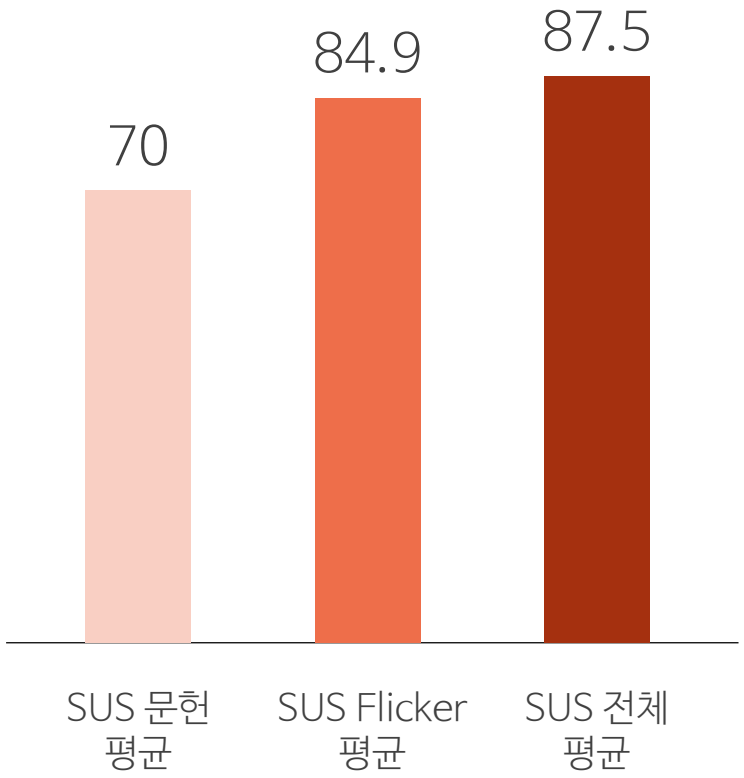
Evaluation: User study



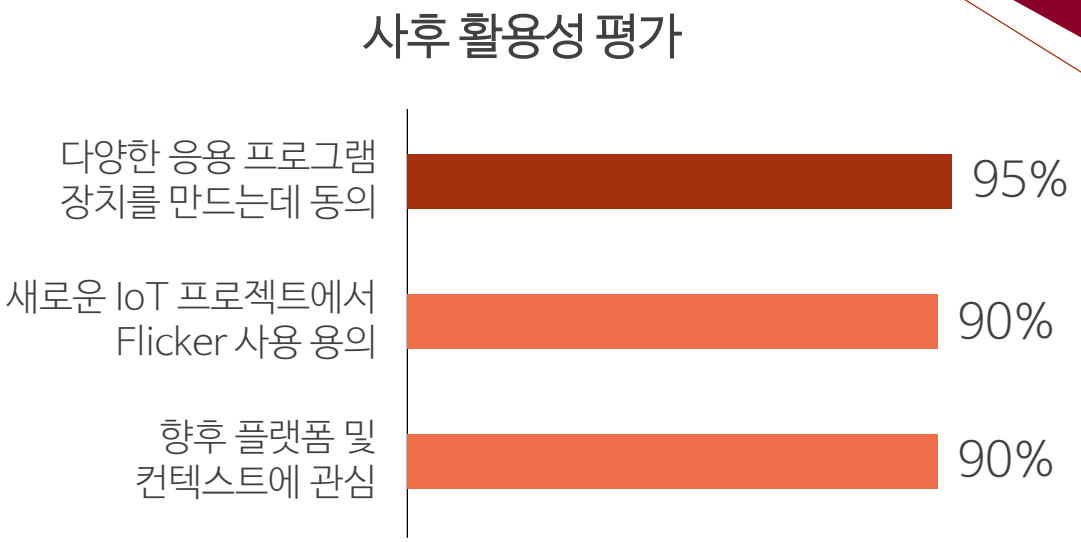
- Responding to programming ability
- Evaluates system and computer knowledge **above average**
- Evaluates **below average** for embedded platforms

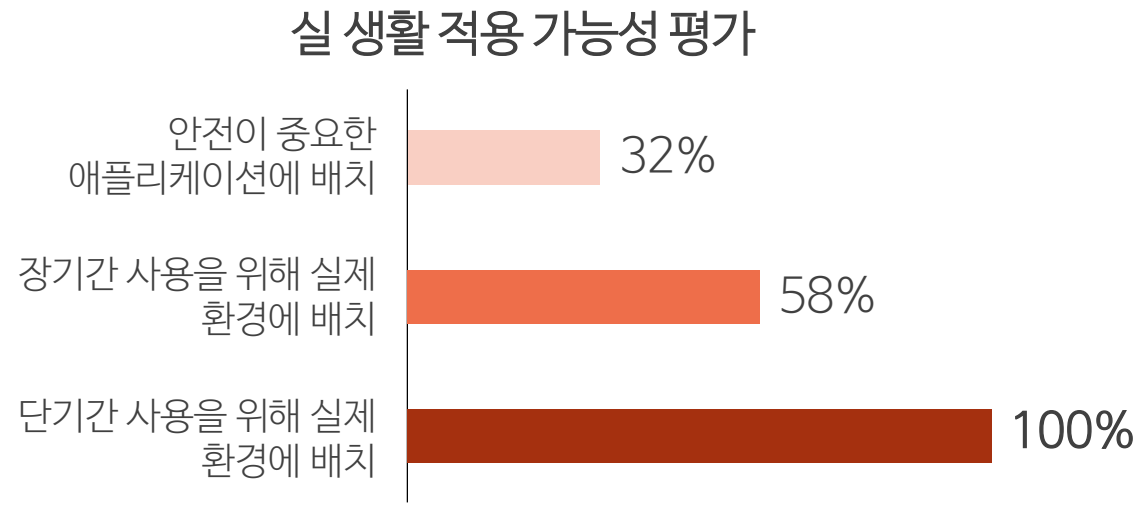
- 5명의 최소 표본 크기를 넘어서 안정적인 규모를 넘어섰다고 말함

Evaluation: User study



※ 19명의 Flicker에 대한 평가와 SUS 평균





- 새로운 플랫폼이기에 유사 솔루션과의 비교가 어렵다고 말함

KOREA UNIVERSITY

Evaluation: Case study

- Flickers do not support multiple harvesters, so hardware changes are required
- Application
  - Prototyping of Plant Bed Greenhouse Monitoring System for Water Conservation
  - Sensors collect leaf moisture and air pressure, and Harvest collects solar energy and RFID energy
  - Simulate applications on mechanical water supply arms following moisture readings in leaves

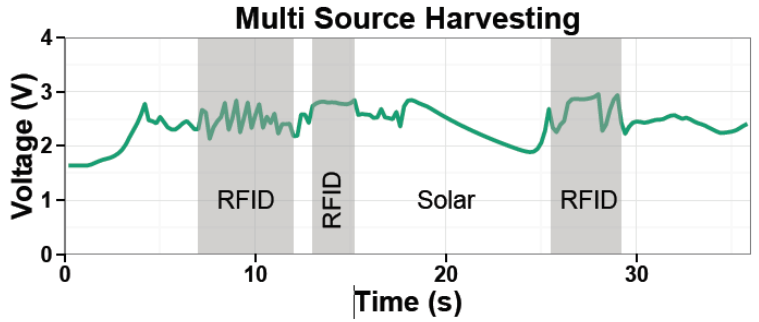


Figure 7: Greenhouse monitoring prototyping case study, capacitor voltage trace shown.

Future work

- Balance modules for reuse
- Tune beyond the limits of changing environments
- Development of measurement criteria for simple operating environment for low-power memory utilization
- Integrating a universal debugging toolchain
- Improvements to peripherals using the eink display, micro SD card or memory IC
- Community for IoT Vision Using Flickers

Conclusion

- An intermittent, battery-free sensor is a new device that defines and enables the IoT vision
- Flickers collect energy without batteries, power-on open source, open hardware platform
- Developers without hardware experience can easily adjust charging threshold values and interrupt routines to match user settings
- Assessment of usefulness in surveys of 19 participants and excellent results
- Flickr supports new battery-free communities and presents new research directions

Critique

- 하드웨어 개발이 어려운 사용자를 위한 자동화된 펌웨어 제공 방식은 급한 사용자를 위해 좋은 솔루션이 될 수 있음

- 설문 대상 19명의 대학교 3-4학년 컴퓨터 공학과 학생을 대상으로 산업 표준 설문을 진행했지만 과연 신뢰성 있는 표본 양인가
  - 과제를 하기 위한 학생들에게 좋은 물건이지 않을까
- 결국 제공하는 모듈이 제한되어 있다면 확장성의 문제가 생길 것

KOREA UNIVERSITY

Thank you

Flicker: Rapid Prototyping for the Batteryless Internet-of-Thing

Sensor Network

컴퓨터공학과 /
2018020727 박휘재