

Developing Real Life, Task Oriented Applications for the Internet of Things

Master's Thesis Presentation

Matheus Amazonas Cabral de Andrade

28th September 2018



Master's Thesis

- Master's in Computing Science (Software Science)
- iCIS's Software Science Department
- Project Supervisor: prof. dr. dr.h.c. ir. M.J. Plasmeijer
- Daily Supervisor: M. Lubbers MSc.
- Second Reader: dr. P.W.M. Koopman



Introduction

- Task-Oriented Programming (TOP)
 - ◊ High level of abstraction
 - ◊ Clean implementation: iTasks
 - ◊ Many platforms
- Internet of Things (IoT)
 - ◊ Microcontrollers
 - ◊ Internet
 - ◊ Ubiquitous
 - ◊ But no TOP



mTask

- Embedded Domain Specific Language
- Motivation
- Two views
 - ◊ Code
 - ◊ Interpret
- The Interpreted View
 - ◊ Motivation
 - ◊ Client
 - ◊ Simulator



mTask Example: Factorial

```
factorial :: (Shared Int) Int -> Main (ByteCode () Stmt)
factorial si i = sds \y=i In lowerSds \x=si In {main =
    IF (y <=. lit 1) (
        pub x :.
        ledOn (lit LED1) :.
        retrn
    ) (
        x =. x *. y :.
        y =. y -. lit 1
    )
}
```



Research Question

Is it possible to develop real-life, IoT applications using mTask? If so, how can the development process be improved? If not, what are the challenges to solve to make it possible?



Application Domain

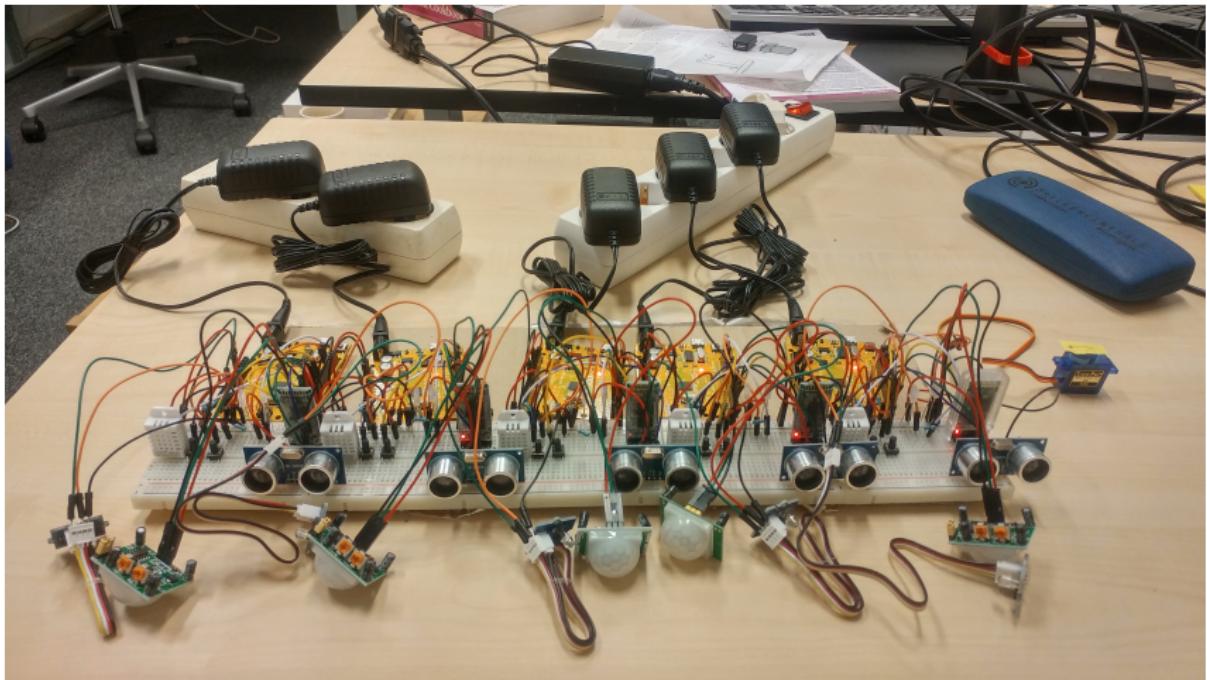
- Selection Criteria
- Home Automation
- Autohouse
 - ◊ Architecture
 - ◊ Tasks
 - ◊ Devices

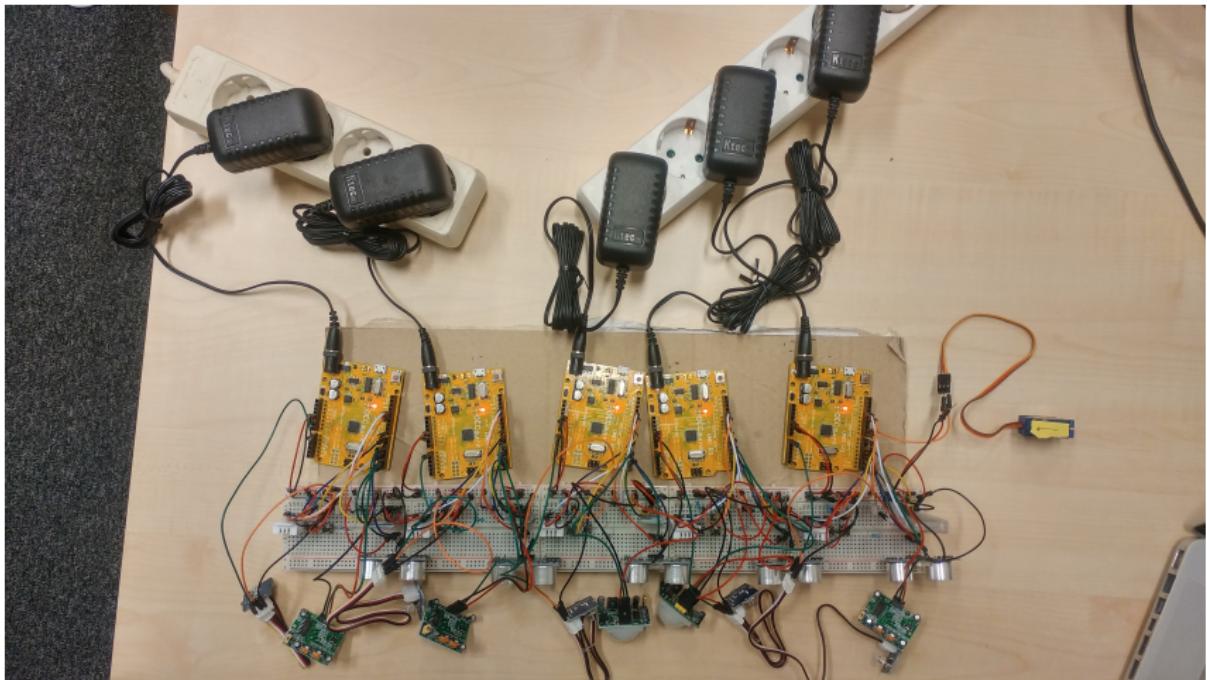


Development Overview

- Simulator on early stages of development
- Wireless communication
- Device deployment







Changes to mTask

- Variables
- New peripherals
- Requirements view
- Device disconnection
- Simulator improvements



Limitations of mTask

- SDSs live forever
- Unwanted SDS update loop
- No task acknowledgment communication
- Floating-point arithmetic



Task Migration

- Device disconnection recogniziton
 - ◊ Using the connection error handler
- Migration strategies
 - ◊ Do not migrate
 - ◊ Same room
 - ◊ Any room
- Migration to compatible devices
 - ◊ Using the Requirements view



Demo



Conclusion

- A real-life application was developed using mTask
- mTask was improved
 - ◊ Changes to mTask
- Future work on mTask
 - ◊ Overcome the limitations found during research
 - ◊ mTask2
 - ◊ Test mTask performance
 - ◊ Test mTask with the Raspberry Pi
- Future work on Autohouse
 - ◊ Task migration with tags
 - ◊ Peripheral sharing



Questions?



Thank you

