

# Ing. Jelle Meeus

## Software Developer

Last update: March 21, 2022

The online version is available at

<https://jellemeeus.github.io/JelleMeeus.github> The

online version is available at

<https://jellemeeus.github.io/JelleMeeus.github>

|        |     |       |     |            |    |      |    |       |    |      |    |
|--------|-----|-------|-----|------------|----|------|----|-------|----|------|----|
| Python | +++ | Git   | +++ | Javascript | ++ | VIM  | ++ | Linux | ++ | Java | ++ |
| CI/CD  | ++  | C/C++ | ++  | PL/SQL     | ++ | 8051 | ++ | React | +  | C#   | +  |
| Docker | +   |       |     |            |    | ASM  |    |       |    |      |    |

### Location

2580 Putte

### Email

[jelle.meeus@hotmail.com](mailto:jelle.meeus@hotmail.com)

### Language

Dutch (native), English (fluent), French (intermediate)

### Driver's license

B

## Education

Master in de industriële wetenschappen, elektronica-ICT 2018

*Katholieke Universiteit Leuven - Campus De Nayer - Sint-Katelijne-Waver*

## Thesis - Continuous Unobtrusive User Authentication Using Gait For Wearable Devices, Utilising Machine Learning Algorithms

[Powerpoint Slides](#) [PDF](#)

In recent years, much research has been done to find new authentication methods that try to avoid explicit input from a user. This techniques use patterns and biometrics from a user to recognize machine learning models. One of these biometrics is the way a person walks. It can be captured by sensors on a smartwatch or smartphone, easily and unobtrusively. The aim of this thesis is to develop a method that is based on an existing barebone implementation. This application consists of a wearable application to record data and a server application to process this data offline. With a study of state-of-the-art recognition of human activity and gait recognition, this implementation was studied, expanded and improved. In this project, a human activity recognition system was placed in cascade with a gait recognition system to design a continuous gait-based authentication model. These systems are traditional machine learning models and use a new feature-extraction technique that is fast and accurate. The new implementation allows data to be captured, to be trained offline on the server and for new data to be evaluated on the wearable. We have explored deep learning, but the traditional approach with manually designed functions performs better.

[Machine Learning](#) [Biometrics](#) [AI](#) [Python](#) [Java](#) [Android](#)

## 8051 Microcontroller Instruction Set IEEE754 32bit Floating-Point Library

Implementation of a IEEE754 Floating-Point library on the aduc832 system platform for 8bit 8052 based systems. The library allows for correct and fast calculation of the 4 basic operations (+, -, \*, /) for two numbers in IEEE754 32b without using MUL/DIV instructions.

[source](#) [ASM](#) [8051](#) [aduc832](#) [IEEE754](#)

## Professional Experience

### Machine Learning Algorithms Student

[OneSpan](#) 4 weeks summer 2017

Developed a demo allowing recording and analysis of gait data on an Android wearable device

[Machine Learning](#) [Biometrics](#) [AI](#) [Python](#) [Java](#) [Android](#)

### Junior Software Engineer Consultant

[Sioux Embedded Systems](#) Nov 18 - Feb 19

I was part of a customer's internal R&D team migrating Version Control Systems (VCS) from Perforce to Git. A large part of my project was to write tools with existing and new features using Python and Gitlab API. Using the same API, I setup a R&D Internal website to provide an overview of projects that updates nightly. I did smaller IT related tasks, such as setting up automated backups of firewall settings and Jenkins configurations. Also, I wrote some automated tests in an in-house testing framework.

[Python](#) [Git](#) [Perforce](#) [Docker](#)

## Hobby projects

[Drawing Cards](#) [demo here](#)

A card drawing web app created with React hosted in github pages. Create and interact with a deck of French-suited SVG rendered cards. You can draw one or multiple cards, shuffle, flip over the deck. All neatly displayed through React with a status bar, menu bar and mouse over interaction.

[source](#) [React](#) [javascript](#)

[Bomberman](#) [play here](#)

Example of a 2D Classic Bomberman game made with unity. You can play as a bomb laying bunny and walk around a maze to reach a carrot and try not blow yourself up in the process.

[source](#) [C#](#) [unity](#)

[Discord Weather Bot](#)

Receive automated weather updates through discord. Allows for lookup of weather forecast at a location with Dark Sky API and OpenWeatherMap API. The pipeline consists of Travis CI, flake8 and pytest.

[source](#) [Python](#) [API](#) [CI](#)