Self-Balancing Robot

簡介

組別：第一組

組長：黃俊瑋（108062308）

組員：黃俊嘉（108062225）

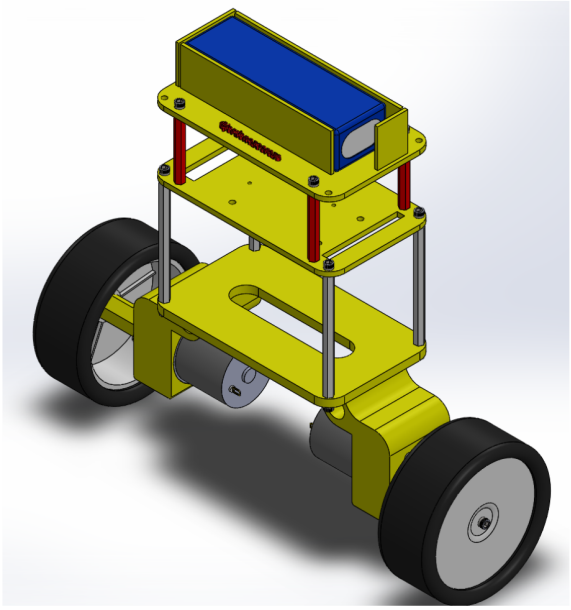
Project Description

簡介

主要會分成 Basic 跟 Advance 來進行，當 Basic 後會開始嘗試 Advance，以避免不成功便成仁的窘境。

Basic: Inverted Pendulum Balancing Robot

使用 FPGA 接收三軸加速度陀螺儀模組的訊號，並控制輪子前後移動以達到平衡的狀態。平衡概念類似於賽格威（Segway）。

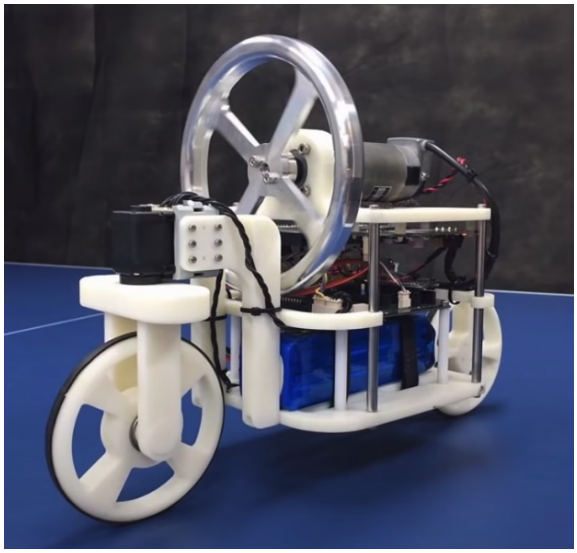


圖片來源：Cerezo, Juan & Morales, Encarnación & Plaza, José. (2019). Control System in Open-Source FPGA for a Self-Balancing Robot. Electronics. 8. 198. 10.3390/electronics8020198.

Advance: Self-Balancing Motorcycle

進一步改良 Basic 時所使用的平衡概念，以旋轉產生反作用力，達到橫向的平衡。加上前後輪馬達，並使用藍芽模組控制整體裝置前進、後退、轉向等行為。

概念來源：
https://www.youtube.com/watch?v=SUVtObDFFWY&feature=youtu.be



圖片來源：同概念來源

設計概念

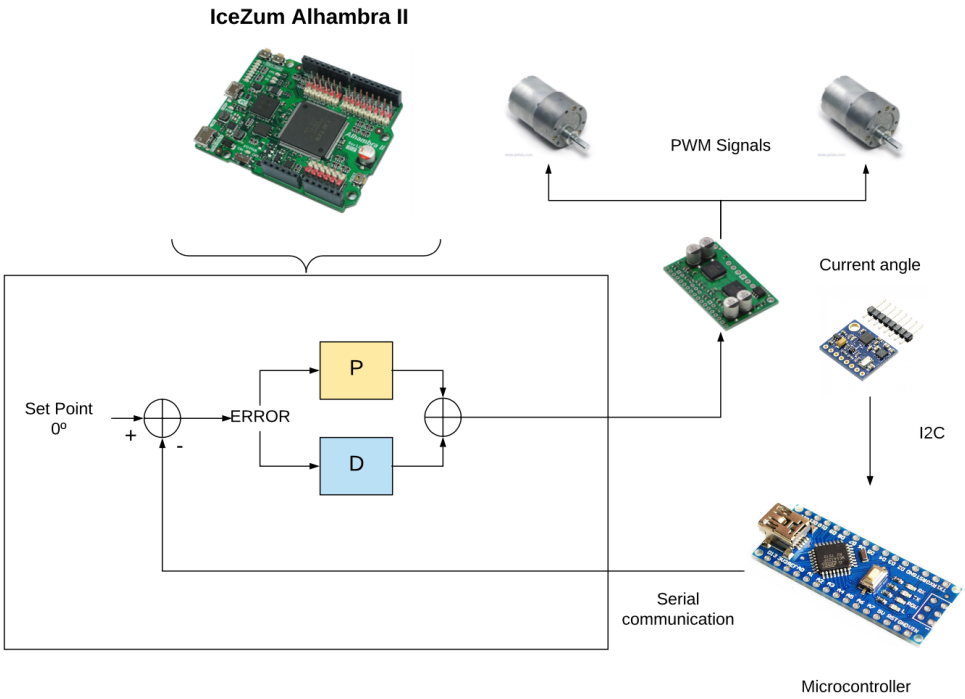
Basic

設計概念取自於 Control System in Open-Source FPGA for a Self-Balancing Robot。

圖片應修改以下幾點（因未知是否合法，故不修改圖片，改為文字描述）：

- 其中將 IceZum Alhambra II 替換成 Basys3 FPGA 板。
- 馬達控制晶片改為 L298N。

1. 使用 MPU6050 感測車體平衡狀態。
2. 使用 Microcontroller（Arduino Nano board）對 MPU6050 的感測訊號進行濾波。
3. 將濾波後的訊號輸入 FPGA 。
4. 計算達到平衡所需的反作用力。
5. 計算達到目標反作用力的馬達方向與轉速。
6. 透過 PID 算法修正旋轉方向與速度的數值。
7. 將修正後的數值傳入馬達控制模組控制馬達，以達到平衡。



圖片來源：Cerezo, Juan & Morales, Encarnación & Plaza, José. (2019). Control System in Open-Source FPGA for a Self-Balancing Robot. Electronics. 8. 198. 10.3390/electronics8020198.

Advance

平衡部分

1. 使用 MPU6050 感測車體平衡狀態。
2. 使用 Microcontroller（Arduino Nano board）對 MPU6050 的感測訊號進行濾波。
3. 將濾波後的訊號輸入 FPGA 。
4. 計算達到平衡所需的反作用力。
5. 計算達到目標反作用力的 Reactoin Wheel 方向與轉速。
6. 透過 PID 算法修正旋轉方向與速度的數值。
7. 使用修正後的數值透過馬達控制模組控制 Reaction Wheel，以達到平衡。

控制部分

1. 使用藍芽晶片接收傳入的控制訊號。
2. 將數值進行調整，確保數值不會過大，導致車速過快難以平衡。
3. 驅動前後輪馬達進行前進、後退、轉向。

預定進行的方法

Basic

- 計算車體重心、力矩...等物理數據。
- 推導所需的反作用力公式，馬達轉向與轉速公式。
- 了解如何使用 Arduino Nano board 接收 MPU6050 的訊號
  - 校準 MPU6050
  - 撰寫濾波演算法
- 了解如何將 Arduino Nano board 訊號傳入 FPGA
  - 撰寫 FPGA 與 Arduino Nano board 的通訊 module
- 了解如何使用 FPGA 控制馬達
  - 撰寫控制馬達的 module
- 計算達到目標反作用力所需的方向與轉速
- 計算經過 PID 演算法後的方向與轉速

Advance

- 計算車體重心、力矩...等物理數據。
- 推導所需的反作用力公式，Reaction Wheel 轉向與轉速公式。
- 計算達到目標反作用力所需的方向與轉速
- 計算經過 PID 演算法後的方向與轉速
- 了解如何使用藍芽模組
  - 撰寫 FPGA 的藍芽通訊 module
  - 撰寫發送控制訊號的軟體

Estimated cost

零件預算表

Name	Price	Usage	Shop	備註
MPU 6050 (GY-521)	100	三軸加速度儀 + 陀螺儀	百年電子露天	
Arduino Nano board (MEGA238P)	100	過濾 MPU 6050 的訊號	百年電子露天	
HC-05	150	藍芽晶片	百年電子露天	
L298N	50	馬達控制模組	百年電子露天	嘗試借用車子的零件
28BYJ-48	100	5V DC 步進馬達 (1)	百年電子露天	
28BYJ-48	100	5V DC 步進馬達 (2)	百年電子露天	
zj327	25	34mm 輪子 (1)	百年電子露天	嘗試借用車子的零件
zj327	25	34mm 輪子 (2)	百年電子露天	嘗試借用車子的零件
冰棒棍	10	拼裝零件用	百年電子露天	
[Advance]				
Reaction Wheel	0	旋轉以提供反作用力	自製	找生活中的東西替代
Gear motor	0	轉動 Reaction Wheel 的馬達	露天	尚未確定使用的型號
L298N	50	馬達控制模組（Reaction Wheel）	百年電子露天	

Schedule

Schedule in Notion (Better UI, much BETTER)：⛑ Self-Balancing Robot

Basic

進度規劃

Aa Name	Date
購買零件	@Dec 15, 2020 → Dec 18, 2020
Research 數學相關資料	@Dec 12, 2020 → Dec 20, 2020
研究如何使用各種控制模組	@Dec 19, 2020 → Dec 25, 2020
撰寫濾波演算法	@Dec 26, 2020 → Dec 28, 2020
撰寫控制馬達 module	@Dec 26, 2020 → Dec 28, 2020
撰寫 FPGA 與 Arduino Nano board 的通訊 module	@Dec 29, 2020 → Jan 2, 2021
校準 MPU6050	@Dec 19, 2020 → Dec 25, 2020
撰寫計算馬達方向與轉速的 module	@Jan 3, 2021 → Jan 9, 2021
測試與調整	@Jan 10, 2021 → Jan 13, 2021
撰寫 PID 演算法的 module	@Jan 3, 2021 → Jan 9, 2021

Advance

如 Basic 進行順利，提早完成，再嘗試 Advance 部分，因此未排入 Schedule。