

HƯỚNG DẪN PHẦN MỀM

PIF EASY PROGRAMMING

(Sử dụng lập trình iBot – môn học Nhập môn về Kỹ Thuật)

1. Giới thiệu phần mềm

PIF Easy Programming là phần mềm được thành viên Câu lạc bộ Sinh viên Nghiên Cứu Khoa Học Khoa Điện – Điện tử (Pay-It-Forward Club) phát triển dựa trên hệ thống mã nguồn mở (Eclipse RCP, IBM GEF and Texas Instruments CCS toolchain).

PIF Easy Programming là phần mềm sử dụng cho sinh viên năm nhất trong môn Nhập môn về kỹ thuật, dùng để lập trình điều khiển robot (iBot).

Với đặc điểm sinh viên năm nhất chưa được học các môn chuyên ngành, chưa biết kiến thức ngôn ngữ lập trình (ví dụ ngôn ngữ C,...), phần mềm được tạo ra với mục tiêu giúp sinh viên năm nhất tiếp cận một hệ thống điện - điện tử hoàn chỉnh theo một cách đơn giản nhưng vẫn tổng quát và toàn diện.

Đặc điểm phần mềm:

- + PIF Easy Programming lập trình cho vi điều khiển MSP430G2553 của TI (Texas Instruments).
- + Giao diện kéo – thả thân thiện.
- + Sinh viên “lập trình” bằng việc nối các block lại với nhau (trung tự Matlab/Simulink, Labview).
- + Một chương trình sẽ được xây dựng bằng việc kết nối các block → sinh viên có cái nhìn tổng quan về hệ thống, hiểu được trình tự vận hành mà không cần phải biết sâu về cấu trúc vi điều khiển.
- + *Thao tác biên dịch – đổ code vào KIT đơn giản:* Chỉ cần nhấn 1 nút RUN trên thanh công cụ thì chương trình (tạo bằng các block) được dịch ra ngôn ngữ C và tự động nạp vào KIT Robot mà không cần phải link sang bất kỳ phần mềm nào khác (như CCS – trình biên dịch cho vi điều khiển sử dụng trong iBot).

+ Chương trình tạo ra một file mã nguồn C (main.c) để phục vụ nhu cầu xem lại code khi cần.

2. Cài đặt

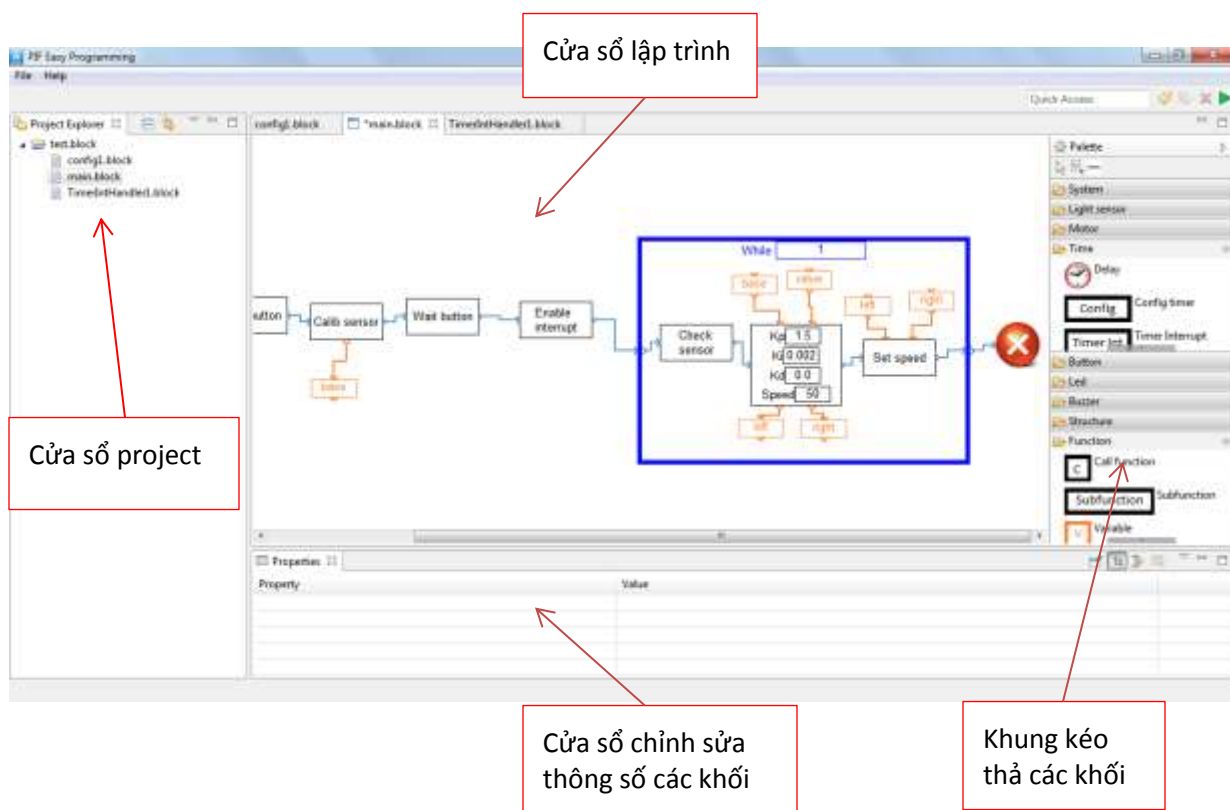
2.1. Yêu cầu hệ thống

- Win XP, Win 7, Win 8 32/64 bit.

2.2. Hướng dẫn cài đặt

- Phần mềm không cần cài đặt.

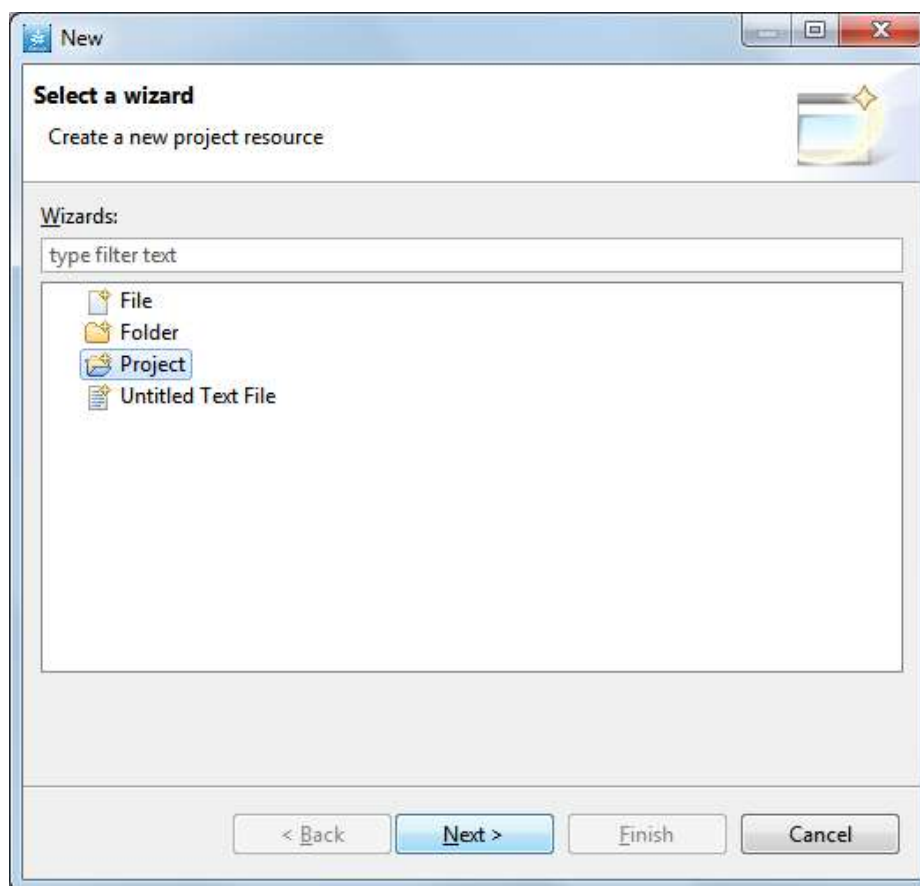
3. Giao diện chương trình



4. Các thao tác cơ bản

4.1. *Tạo project mới*

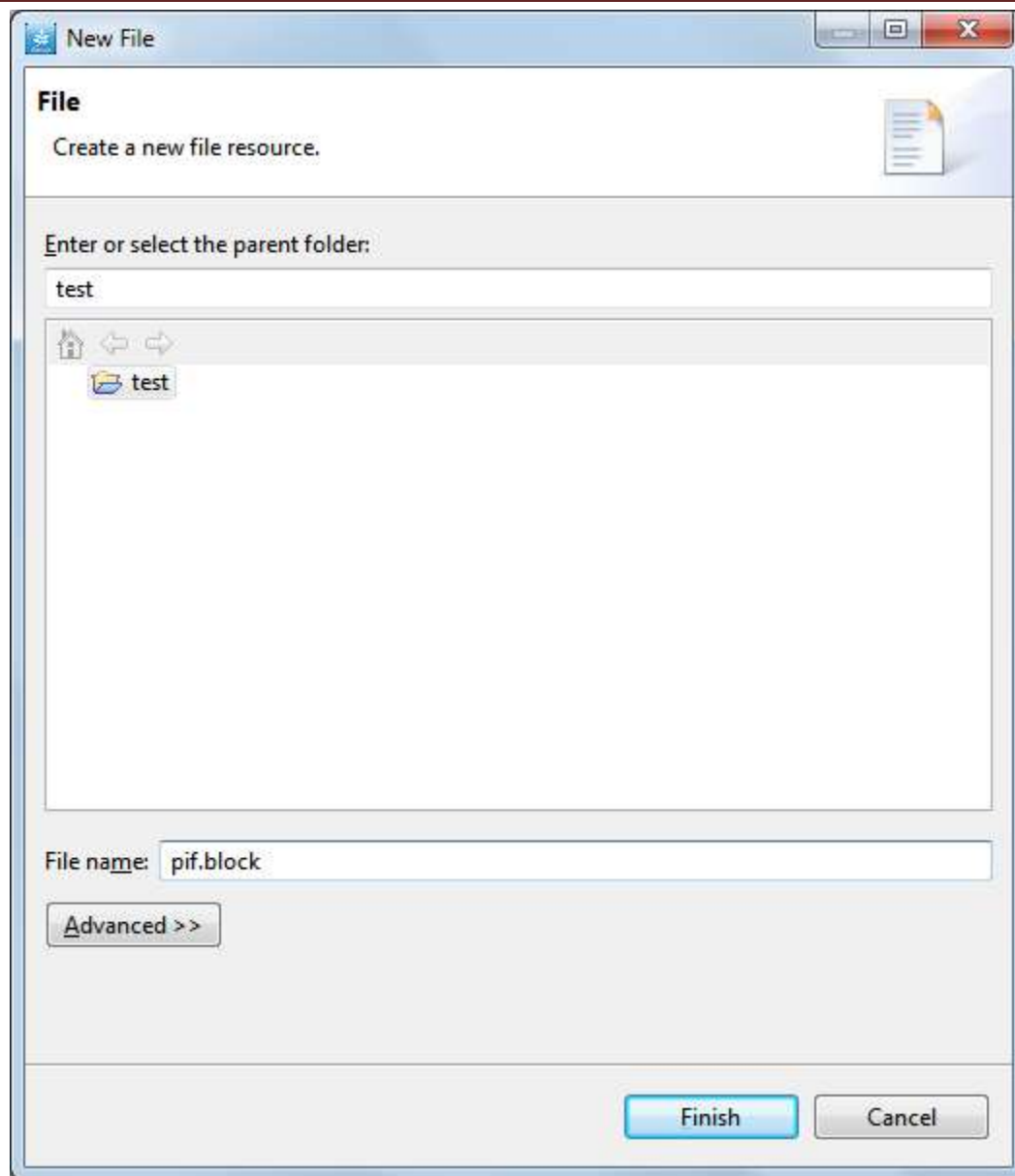
- Chọn menu File-> New, một cửa sổ xuất hiện.



- Chọn Project, nhấn Next.
- Nhập tên project và nhấn Finish.

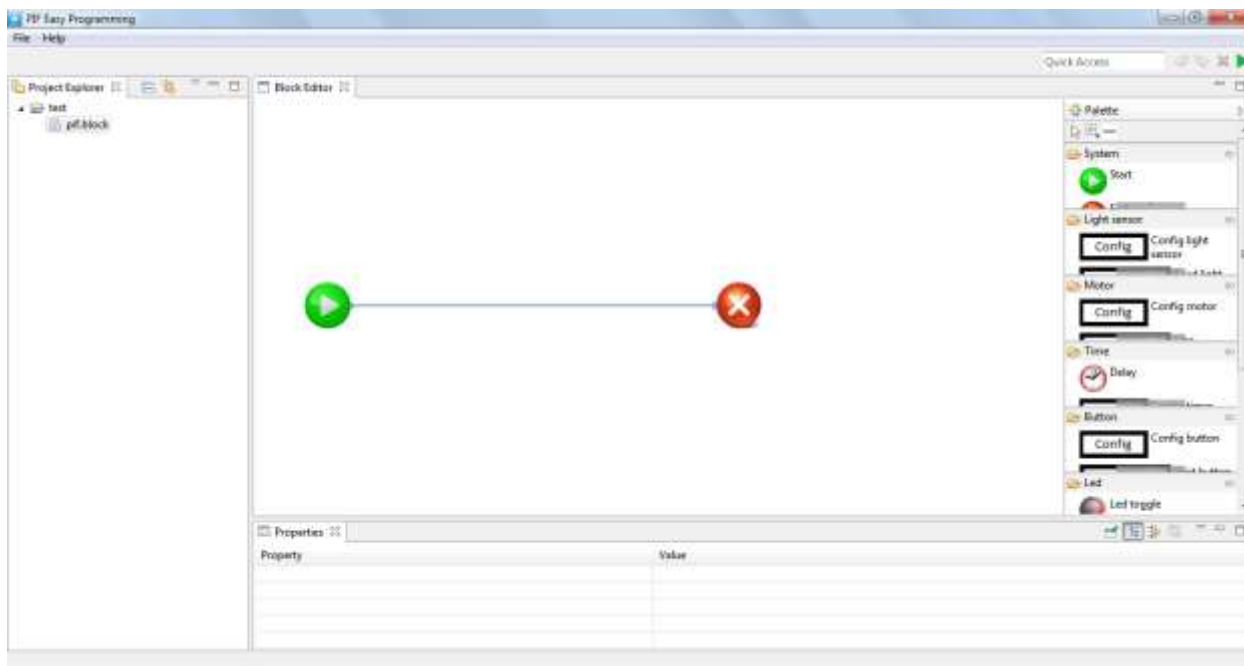
4.2. *Tạo file mới*

- Click phải vào tên project trong cửa sổ Project Explorer, chọn New->File, một cửa sổ xuất hiện.



- Chọn project để chứa file trong danh sách các project, nhập tên file với phần mở rộng *.block*.
- ❖ **Lưu ý:** với mỗi project, bạn chỉ tự tạo mới một file với *.block* lúc ban đầu, các file *.block* khác sẽ được chương trình tự động tạo ra khi bạn kéo thả các khối vào cửa sổ lập trình.


- Khi tạo project mới thành công, một cửa sổ lập trình sẽ xuất hiện với 2 khối Start, End như hình sau:

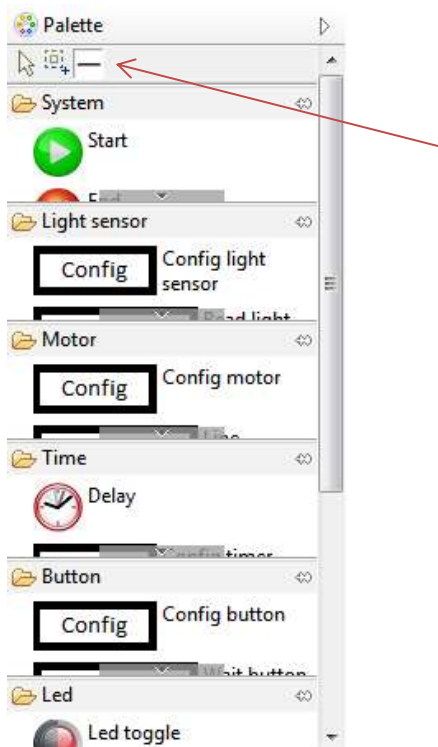


4.3. Thêm một khối vào cửa sổ lập trình

- Kéo khối tương ứng trong khung chứa các khối vào cửa sổ lập trình.

4.4. Kết nối các khối

- Nhấn vào biểu tượng  ở khung bên trái cửa sổ lập trình và nối ngõ ra của khối này vào ngõ vào của khối kia.



- Nếu muốn nối lại đoạn dây, ta click vào và di chuyển đầu hay cuối đoạn dây đến khối khác.

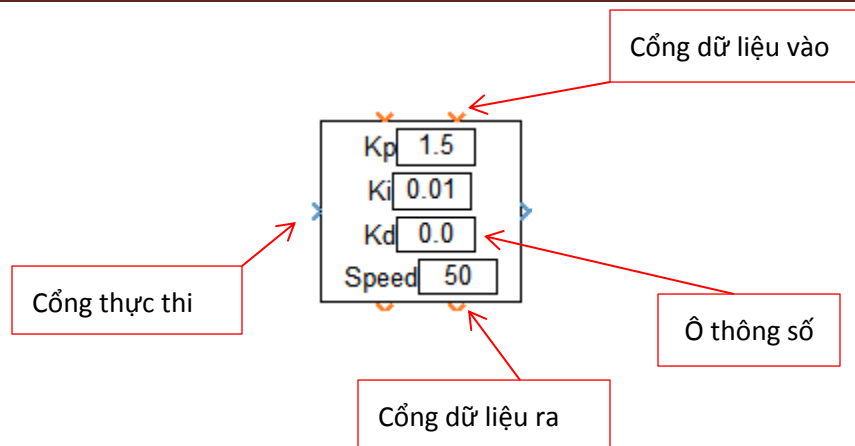
4.5. Nạp chương trình cho robot

- Bạn bấm vào biểu tượng Run hình mũi tên màu xanh trên thanh công cụ hay nhấn phím tắt Ctrl+R.



5. Các khối chức năng

- Tổng quát, một khối chức năng bao gồm: cổng thực thi (màu xanh dương, quyết định thứ tự thực hiện chương trình), cổng dữ liệu (màu cam, dữ liệu vào/ra của khối) và ô thông số (quyết định cách khối được thực thi).



- Cổng dữ liệu chỉ kết nối với khối Variable trong nhóm Function.
- Các thông số có thể được thay đổi trực tiếp bằng cách click chuột vào ô thông số hay sử dụng cửa sổ Properties ở bên dưới cửa sổ lập trình.

Các nhóm khối chức năng trong chương trình:

5.1. Nhóm System

Khối *Start*: bắt đầu chương trình chính, bạn chỉ có thể đặt một khối Start block trong chương trình.

Khối *Stop*: kết thúc chương trình chính, bạn chỉ có thể đặt một khối Stop block trong chương trình.

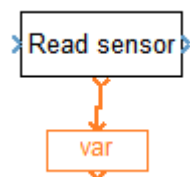
Khối *Config*: cấu hình các module của robot.

Khối *Enable Interrupt*: cho phép ngắt.

5.2. Nhóm Light sensor

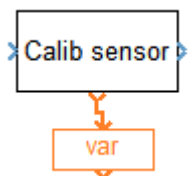
Khối *Config*: cấu hình module cảm biến quang.

Khối *Read Sensor*: đọc cảm biến quang, giá trị của cảm biến được chứa trong khối Variable kết nối với ngõ dữ liệu ra.



Khối *Check sensor*: Kiểm tra cảm biến đã được đọc xong chưa, thường đứng sau khối ReadSensor.

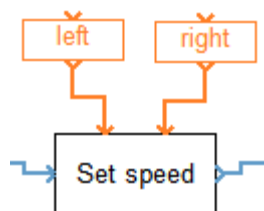
Khối *Calib sensor*: lấy giá trị ban đầu của cảm biến.



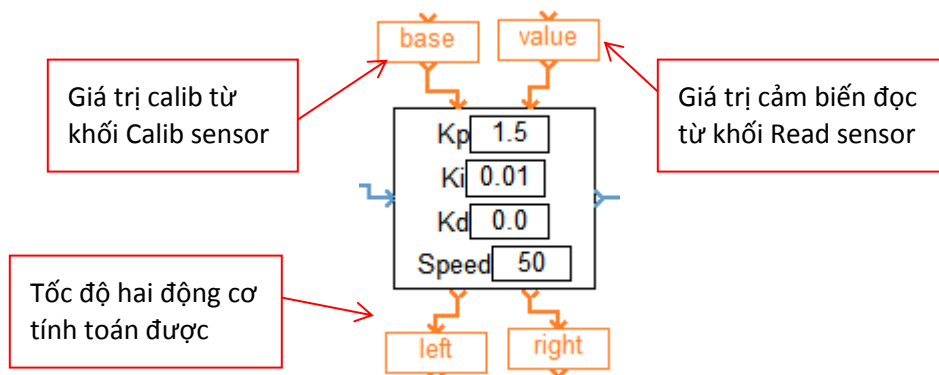
5.3. Nhóm Motor

Khối *Config*: cấu hình module điều khiển động cơ.

Khối *Set Speed*: đặt tốc độ 2 động cơ.



Khối *Follow Line*: thực hiện thuật toán bám line.



5.4. Nhóm Time

Khởi *Config*: cấu hình module timer, giá trị trong ô thông số là thời gian giữa hai lần hàm ngắt được gọi.

Khởi *Delay*: delay robot một khoảng thời gian tính theo đơn vị *ms*.

Khởi *Timer Interrupt*: double click vào để viết chương trình ngắt cho timer.

5.5. Nhóm Button

Khởi *Config*: cấu hình module nút nhấn

Khởi *Wait Button*: chờ người dùng bấm nút bên trái hay phải.

5.6. Nhóm Led

Khởi *Config*: cấu hình module led.

Khởi *Led Toggle*: đảo trạng thái led.

Khởi *Led On*: bật led trái hay phải.

Khởi *Led Off*: tắt led trái hay phải.

5.7. Nhóm Buzzer

Khởi *Config*: cấu hình module buzzer.

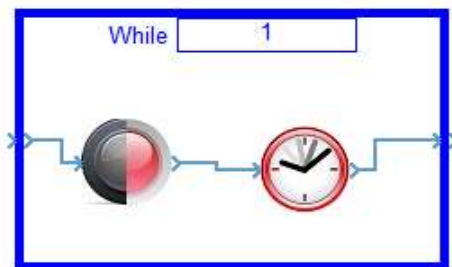
Khởi *Buzzer On*: bật buzzer.

Khởi *Buzzer Off*: tắt buzzer.

5.8. Nhóm Structure

Khởi **While**: các khối đặt trong khối While sẽ được thực hiện lặp lại vô tận.

Ví dụ: đoạn chương trình sau sẽ đảo trạng thái led sau mỗi khoảng thời gian xác định bởi khối delay.

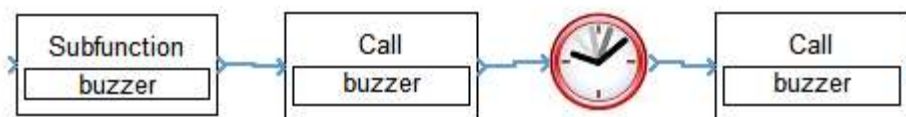


5.9. Nhóm Function

- Khối *Subfunction*: double click vào để tạo chương trình con, thường dùng với khối Call function.
- Khối *Call function*: gọi chương trình con, dùng với khối Subfunction.

Ví dụ:

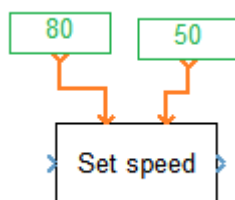
Đoạn chương trình sau tạo chương trình con mang tên buzzer và gọi chương trình này, dừng một khoảng thời gian và gọi lại chương trình buzzer.



Chương trình con buzzer sẽ bật buzzer một khoảng thời gian rồi tắt đi.

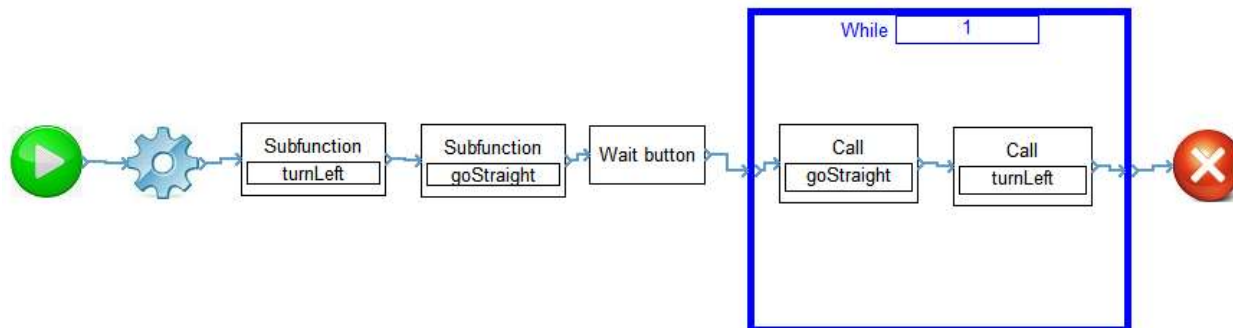


- Khối *Variable*: chứa dữ liệu vào/ra của các block, kết nối với cổng dữ liệu của các khối.
- Khối *Constant*: chứa dữ liệu hằng số, dùng kết nối với cổng dữ liệu vào của khối Set Speed.

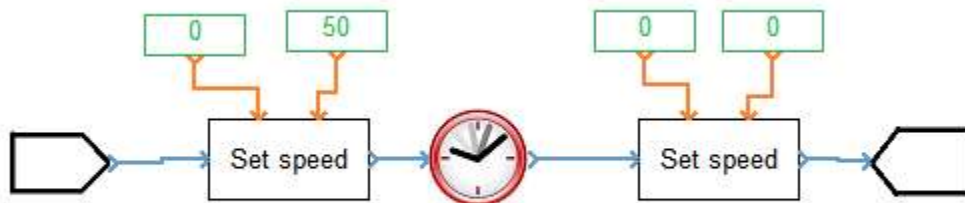


6. Ví dụ mẫu

Ví dụ 1: Chương trình cho iBot đi hình vuông

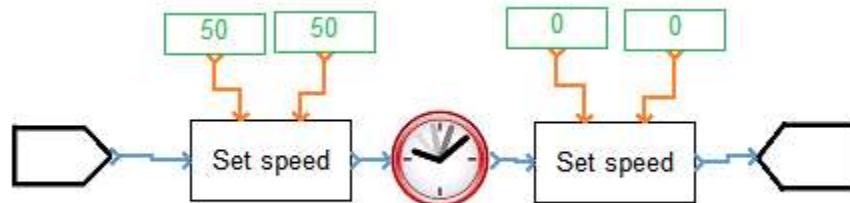


Chương trình turnLeft

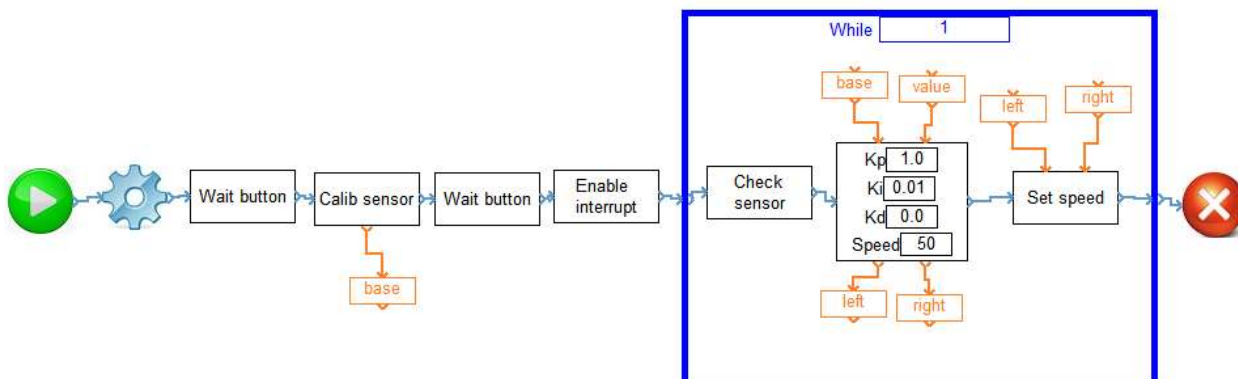


Chọn thời gian trễ (delay) phù hợp để robot quay góc 90 độ.

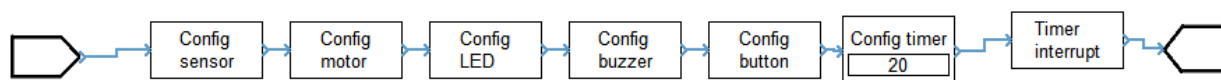
Chương trình goStraight



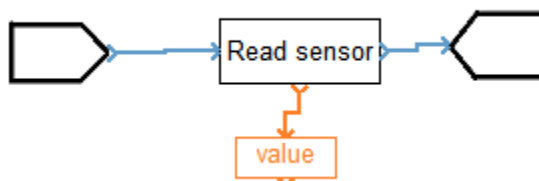
Ví dụ 2: Chương trình dò line



Trong khối Config, bạn thêm vào hai khối *Timer Config* và *Timer Interrupt*.



Khối *Timer Interrupt*



7. Các lỗi có thể gặp và nguyên nhân

7.1. *There are at least two Start blocks.*

Chương trình có nhiều hơn một khối bắt đầu.

7.2. *There is no Start block.*

Chương trình không chứa khối Start block.

7.3. *Control flow is not continuous.*

Công thực thi của các khối chưa được kết nối liên tục với nhau.

7.4. *Data port is not connected yet.*

Cổng dữ liệu của khối chưa được kết nối.

7.5. *Variable has inconsistent type, Constant and data port have inconsistent data type.*

Kiểu dữ liệu không khớp. Ngõ vào và ra của khối PID có kiểu dữ liệu khác nhau, không thể chứa trong cùng một Var block, bạn hãy đổi tên Var block.

7.6. *Function ... is called but is not declared yet.*

Khối Call function gọi chương trình con chưa được tạo, sử dụng khối Subfunction để tạo chương trình con.

7.7. *There are at least two interrupt blocks from one module.*

Có hai khối Timer Interrupt block trong chương trình, bạn hãy xóa đi một khối.

8. Download phần mềm, thông tin tác giả:

Java Runtime Environment (JRE)

Cần cài đặt JRE để chạy chương trình PIF Easy Programming.

www.java.com/en/download/manual.jsp