Ngôn ngữ C

Ngôn ngữ C trong lập trình arduino

void setup( ) {}

//hàm này chỉ chạy 1 lần duy nhất khi cấp điện hoặc sau bấm nút reset

//void => cấu trúc khai báo hàm: hàm ko trả về dữ liệu

//setup : đây là tên hàm

//( ) để chứa các tham số, ở hàm setup thì ko có tham số

void loop( ){. . .} // Hàm này chạy xong tự chạy lại mãi mãi

Hàm trả về dữ liệu: cần có kiểu khác với void

kiểu_dữ_liệu Tên_hàm(){

Thân hàm

Chứa lệnh(s) **return** giá_trị;

}

Trong C có các kiểu dữ liệu sau:

1. KIỂU SỐ NGUYÊN: char, int, long

: số **có dấu (bít đầu là bít dấu)**

QUY ƯỚC:

bít đầu =1: SỐ ÂM

bít đầu=0: SỐ DƯƠNG

dãy biểu diễn số âm N + dãy nhị phân biểu diễn +N => "0" trong không gian 8 bít

sv cn sn A: can 10 lít đi mua rượu, đầy can. Thêm 1 lít, đổ vào can 10 lít. =>tràn mất
dãy nhị phân biểu diễn +N = 1 0000 0000- dãy biểu diễn số âm N

a. **char**: số nguyên 1 byte: -127 … +127
b. **int**: 2byte. Miền giá trị -2^15=>+(2^15)-1
c. **long**: 4 byte=32 bít; biểu diễn 2^32 giá trị khác nhau
   miền -2^31 =>+2^{31}-1
   -2^{31} ….,-2, -1 : có 2^31 giá trị
   0 (1 giá trị)
   1,2,3,4…2^31-1: có 2^31-1 giá trị
   ---------- 2(2^31)=2^32

2. SỐ NGUYÊN KHÔNG DẤU

(toàn bộ dãy nhị phân đều là số dương)

a. **unsigned** char : kiểu số nguyên 1 byte ko dấu: 0…255
b. **unsigned** int : kiểu số nguyên 2 byte ko dấu: 0…2^16-1
c. **unsigned** long: kiểu số nguyên 4 byte ko dấu: 0…2^32-1

3. KIỂU LOGIC BOOL (kiểu đúng sai)

bool: chỉ nhận 2 giá trị: **true** hoặc **false**

khi giá trị gán cho 1 biến ko cùng kiểu với khai báo của biến

thì ÉP KIỂU TỰ ĐỘNG => KQ VỀ KIỂU KHAI BÁO

ÉP KIỂU CHỦ ĐỘNG: (kiểu mong muốn)biểu thức hoặc biến

int a=5;

int b=3.14; //ep kieu tu dong b=3

//

char kiểu kí tự: gán cho nó kí tự:
char c = ‘A’;//gán kí tự A cho biến c || c nhận mã ascii của kí tự A

thử lại
200= 11001000 (bít cao==1 => Âm)= - 56

56 = 00111000 (số dương)
------1 0000 0000---- ==0 trong ko gian 8 bit

kiểu dữ liệu so nguyen: **char, int, long**

ko dấu: thêm tiền tố **unsigned**

4. KIỂU SỐ THỰC:

double, float

kq pheps chia => so thuc: TỬ HOẶC MẪU HOẶC CẢ 2: LÀ SỐ THỰC

NẾU TỬ VÀ MẪU ĐỀU LÀ SỐ NGUYÊN => KQ CHIA LÀ SỐ NGUYÊN

ÉP CHỦ ĐỘNG MẪU OR TỦ => SỐ THỰC

KHÓ KHĂN KHI LẬP TRÌNH: THUẬT TOÁN

NGÔN NGỮ LẬP TRÌNH: EASY

PHẦN CỨNG: CÓ NGUYÊN TẮC, NGUYÊN LÝ TỪ NSX

THIẾT BỊ NGOẠI VI: NGUYÊN TẮC HOẠT ĐỘNG VỀ ĐIỆN FIX SẴN.

TẠO RA 1 HỆ THỐNG TỰ ĐỘNG HOÁ/IOT: THÔNG MINH,…, khó ở tư duy logic xử lý vấn đề.

5. Kiểu du lieu, ép kiểu:

auto, by hand.

Nhúng: Số

Đọc từ cảm biến được số => biến để lưu với kiểu phù hợp

Có thư viện làm hộ rồi, tìm đúng thông tin của cảm biến, tra cứu thư viện, kéo thư viện về, xem vd mẫu.

Hiển thị: led7 thanh, led matrix, oled, LCD16x2,… màn hình đặc tru3wng của lập jtrinfh nhugs: ko PRO như Monitor. Mà nó rất hạn chế, tìm đặc trưng, xuất dữ liệu vào đúng đặc trưng đó

6. Một số toán tử trong C

+ - * /

% chia lấy dư

So sánh: >, <, >=, <=, ==, !=

Logic: && AND, || or , ! NOT

Toán tử bít:

& AND BIT theo từng cặp

| OR BIT theo từng cặp

int a=5, b=4

XOR bit theo từng cặp (+ ko nhớ)

Tổng kết buổi học 1:

1. Kiểu dữ liệu:

Số nguyên: min, max, unsigned +kdl => kiểu ko dấu

Số thực: float, double

Kiểu bool: đúng sai, ép tự động số khác 0 là đúng. Đúng=>1, sai->0

2. toán tử: toán học, chia dư %, so sánh == != , toán tử LOGIC && || !

Toán tử bit (làm việc theo cặp bít): &, | ^

7. cấu trúc điều khiển: if(biểu thức bool)L1;

ekse L2;
if có thể lồng nhau.

8. Toán tử ?: (toán tử 3 ngôi)

btbool?value_khi true:value khi false

Vd is_snt: tạo hàm trả về true/false => hàm đ.n ntn

Hàm có trả về dữ liệu

bool is_snt(int n){

…return true

…return false

}.

Code viết trước xoá bớt đi: đọc cảm biến DHT11

Code sau khi xoá: quá dễ: khai báo tv, khởi tạo biến dùng chân D4

Setup(): dht.begin()

Loop(): int t=dht.readTemp..()

//soi vào thư viện .h rồi, .cpp rồi hơn, nhiều toán tử bít, ngủ 1ms, 100ms,….
Đọc 40 bit……. Phức tạp: do nhà sx nó đẩy data như vậy

Thư viện chắc chắn phải như thế mới đọc được

Thư viện nó làm hộ rồi, trình nn C hiểu hết.

//Mọi cảm biến trên đời: nsx đều mong các nhà phát triển dùng sensor này. Bán đc hàng, cách sử dụng phải public, code mẫu (thư viện)

C khó, dùng thông qua thư viện, exam (vd mẫu) thì lại rất dễ.