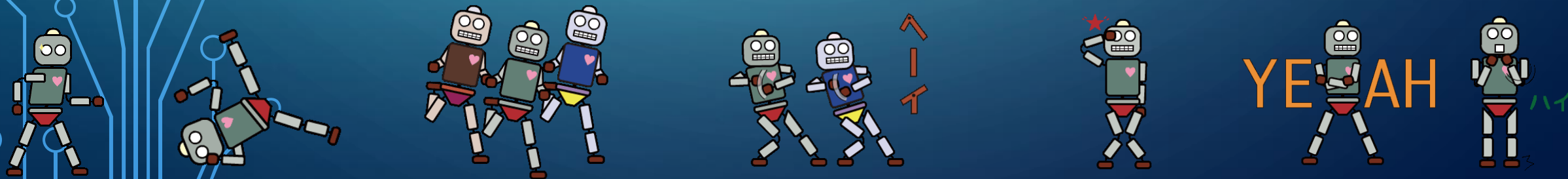




TRƯỜNG CAO ĐẲNG LÝ TỰ TRỌNG TP.HCM KHOA ĐIỆN – ĐIỆN TỬ

HỌC PHẦN

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG



Chương 2: LẬP TRÌNH C TRÊN ARDUINO IDE

1. Lập trình Arduino (2 tiết)
2. Các kiểu dữ liệu
3. Chuyển đổi kiểu dữ liệu (2 tiết)
4. Hàm
5. Lệnh điều kiện
6. Lệnh vòng lặp (2 tiết)
7. Các phép tính toán
8. Xử lý chuỗi ký tự

Chương 2: LẬP TRÌNH C TRÊN ARDUINO IDE

6. Lệnh vòng lặp

Vòng lặp sẽ thực hiện nhiều lần (làm đi làm lại)
một công việc có một tính chất chung nào đó

6.1. while

Vòng lặp **while** là dạng vòng lặp theo điều kiện,
Người lập trình chủ động quản lý lúc nào thì vòng lặp ngừng lặp
(*nhưng sẽ không biết trước số lần lặp*)

Cú pháp:

```
while ([điều_kiện]) {  
  //các câu lệnh;  
}
```

Chương 2: LẬP TRÌNH C TRÊN ARDUINO IDE

6. Lệnh vòng lặp

6.1. while

Ví dụ: `while ([điều_kiện]) {`
 //các câu lệnh;
 `}`

Chương 2: LẬP TRÌNH C TRÊN ARDUINO IDE

6. Lệnh vòng lặp

6.2. for

Vòng lặp **for** là dạng vòng lặp theo điều kiện biết trước số lần lặp

Cú pháp:

```
for ([biến_chạy] = [giá_trị_bắt_đầu];[điều_kiện];[bước_chạy])  
{  
  //câu_lệnh  
}
```

Chương 2: LẬP TRÌNH C TRÊN ARDUINO IDE

6. Lệnh vòng lặp

6.2. for

Ví dụ:

```
for ([biến_chạy] = [giá_trị_bắt_đầu];[điều_kiện];[bước_chạy])  
{  
  //câu_lệnh  
}
```

Chương 2: LẬP TRÌNH C TRÊN ARDUINO IDE

6. Lệnh vòng lặp

6.3. break

break là lệnh có chức năng dừng ngay lập tức một vòng lặp (for, while) chứa nó trong đó.

Khi dừng vòng lặp, tất cả những lệnh phía sau break và ở trong vòng lặp chịu ảnh hưởng của nó sẽ bị bỏ qua.

Ví dụ:

```
int a = 0;
```

```
while (true) {  
    if (a == 5) break;  
    a = a + 1;  
}
```

```
//a = 5
```


Chương 2: LẬP TRÌNH C TRÊN ARDUINO IDE

6. Lệnh vòng lặp

6.3. break

Ví dụ:

```
int a = 0;
```

```
while (true) {  
    while (true) {  
        a++;  
        if (a > 5) break;  
    }  
    a++;  
    if (a > 100) break;  
}
```

//a = 101

Chương 2: LẬP TRÌNH C TRÊN ARDUINO IDE

6. Lệnh vòng lặp

6.3. continue

continue là lệnh có chức năng bỏ qua một chu kì lặp trong một vòng lặp (for, while) chứa nó trong đó.

Khi gọi lệnh continue, những lệnh sau nó và ở trong cùng vòng lặp với nó sẽ bị bỏ qua để thực hiện những chu kì lặp kế tiếp.

Ví dụ:

```
int a = 0;
int i = 0;
while (i < 10) {
    i = i + 1;
    continue;
    a = 1;
}
```

//a vẫn bằng 0

Chương 2: LẬP TRÌNH C TRÊN ARDUINO IDE

6. Lệnh vòng lặp

6.4. goto Lệnh **goto** sẽ chuyển đến một nhãn đã được định trước

Lệnh **goto** thường được dùng trong trường hợp đang dùng nhiều vòng lặp quá và muốn thoát khỏi các vòng lặp đó một cách nhanh chóng

Ví dụ:

```
for(byte r = 0; r < 255; r++){  
    for(byte g = 255; g > -1; g--){  
        for(byte b = 0; b < 255; b++){  
            if (analogRead(0) > 250){ goto thoaatra;}  
            //thêm nhiều câu lệnh nữa  
        }  
    }  
}  
thoaatra:
```

Chương 2: LẬP TRÌNH C TRÊN ARDUINO IDE

* Arduino Serial Monitor

Serial Monitor là thành phần của Arduino IDE, giúp board mạch Arduino và máy tính có thể gửi và nhận dữ liệu với nhau qua cổng USB

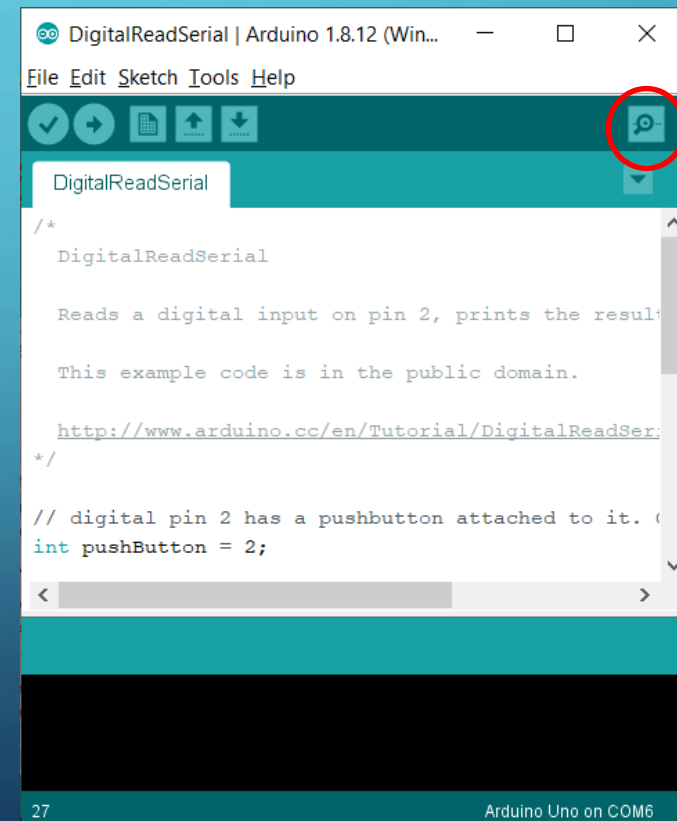
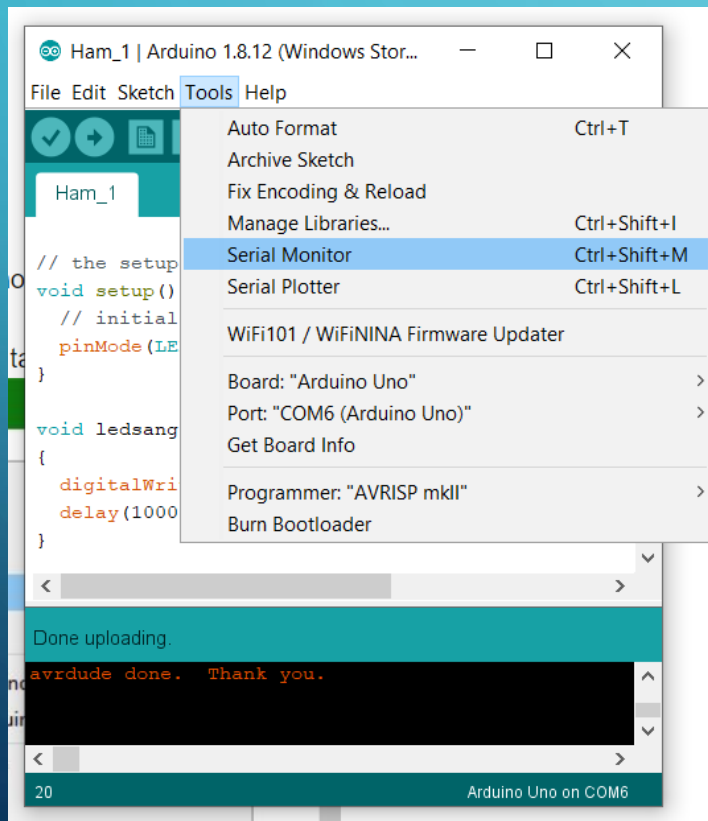
Thông qua Serial Monitor, có thể hiển thị và nhập dữ liệu cho chương trình đang chạy trên board mạch Arduino.

Chương 2: LẬP TRÌNH C TRÊN ARDUINO IDE

* Arduino Serial Monitor

1. Để mở cửa sổ Serial Monitor:

- Cần đảm bảo đã kết nối với board mạch Arduino



Chương 2: LẬP TRÌNH C TRÊN ARDUINO IDE

* Arduino Serial Monitor

2. Để giao tiếp qua **Serial Monitor**:

- Trong hàm ***void setup()*** thiết lập tốc độ kết nối Serial UART bằng hàm ***Serial.begin()***

```
void setup() {  
    // initialize serial communication at 9600 bits per second  
    Serial.begin(9600);  
}
```

Chương 2: LẬP TRÌNH C TRÊN ARDUINO IDE

* Arduino Serial Monitor

3. Để hiển thị nội dung ra màn hình **Serial Monitor**:

dùng hàm **Serial.print()**
hoặc **Serial.println()**

```
void loop()
{
  Serial.print("Hello world\n");
  delay(1000);
}
```


Chương 2: LẬP TRÌNH C TRÊN ARDUINO IDE

* Arduino Serial Monitor

4. Để nhận dữ liệu nhập từ màn hình **Serial Monitor**:

Arduino Uno đã có sẵn bộ đệm UART (buffer) có kích cỡ 64byte. Mỗi lần nhận 1 ký tự thì ký tự này sẽ được tự động chuyển vào bộ đệm. Người dùng muốn đọc dữ liệu chỉ cần làm việc với bộ đệm.

+ Kiểm tra bộ đệm: Trước khi thực hiện thao tác đọc và xử lý dữ liệu của thiết bị khác gửi đến thì cần kiểm tra bộ đệm xem có sẵn sàng chưa bằng hàm **Serial.available()**.

Hàm **Serial.available()** sẽ trả về số kí tự (byte) hiện có trong bộ đệm.

```
if(Serial.available() > 0) {  
  // đọc và xử lý dữ liệu khi có dữ liệu  
}
```


Chương 2: LẬP TRÌNH C TRÊN ARDUINO IDE

* Arduino Serial Monitor

4. Để nhận dữ liệu nhập từ màn hình **Serial Monitor**:

+ **Đọc dữ liệu**: Để đọc 1 byte từ bộ đệm, sử dụng hàm **Serial.read()**

Trong thực tế, hầu hết là cần **đọc 1 chuỗi ký tự**.
Có thể đọc từng ký tự, sau đó ghép lại thành 1 chuỗi
hoặc sử dụng hàm **Serial.readString()**

+ **Bộ đệm UART** sẽ mất dữ liệu ngay sau khi sử dụng lệnh đọc
→ Nên đọc và lưu ra 1 biến và làm việc với biến đó

Chương 2: LẬP TRÌNH C TRÊN ARDUINO IDE

* Arduino Serial Monitor

4. Để nhận dữ liệu nhập từ màn hình **Serial Monitor**:

Ví dụ:

```
String text;  
void setup()  
{  
  Serial.begin(9600);  
  while (!Serial) {}  
  Serial.println("bam on de bat led /  
off de tat led");  
  pinMode(13, OUTPUT);  
  digitalWrite(13, LOW);  
}
```

```
void loop()  
{  
  if (Serial.available() > 0){  
    text = Serial.readStringUntil('\n');  
    text.trim();  
    if(text == "on")  
      digitalWrite(13,HIGH);  
    if(text == "off")  
      digitalWrite(13,LOW);  
  }  
}
```

Chương 2: LẬP TRÌNH C TRÊN ARDUINO IDE

7. Các phép tính toán

7.1. Cộng, trừ, nhân, chia

STT	Toán tử	Tên	Ví dụ	Kết quả
1	+	Cộng	$3 + 4$	7
2	-	Trừ	$10.5 - 3$	7.5
3	*	Nhân	$3 * 2$	6
4	/	Chia	$6 / 3$	2
5	%	Chia lấy dư	$8 \% 3$	2

Chương 2: LẬP TRÌNH C TRÊN ARDUINO IDE

7. Các phép tính toán

7.1. Cộng, trừ, nhân, chia

- Lưu ý:**
1. Nếu cả hai toán hạng là số nguyên thì kết quả sẽ là một số nguyên
 2. Nếu một hoặc cả hai toán hạng là số thực thì kết quả sẽ là một số thực
 3. Khi cả hai toán hạng của phép chia là số nguyên thì kết quả là phần nguyên của thương số
- Để lấy được đầy đủ thương số, cần chuyển kiểu dữ liệu sang số thực

Chương 2: LẬP TRÌNH C TRÊN ARDUINO IDE

7. Các phép tính toán

7.2. Tăng, giảm giá trị thêm 1 đơn vị

```
void setup() {  
  Serial.begin(9600);  
}
```

```
void loop() {  
  int i, k;
```

```
    i = 5; k = i++;  
    Serial.print("i= ");  
    Serial.print(i);  
    Serial.print("\t");  
    Serial.print("k= ");  
    Serial.println(k);
```

```
    i = 5; k = ++i;  
    Serial.print("i= ");  
    Serial.print(i);  
    Serial.print("\t");  
    Serial.print("k= ");  
    Serial.println(k);
```

```
    i = 5; k = i--;  
    Serial.print("i= ");  
    Serial.print(i);  
    Serial.print("\t");  
    Serial.print("k= ");  
    Serial.println(k);
```

```
    i = 5; k = --i;  
    Serial.print("i= ");  
    Serial.print(i);  
    Serial.print("\t");  
    Serial.print("k= ");  
    Serial.println(k);  
}
```

Chương 2: LẬP TRÌNH C TRÊN ARDUINO IDE

7. Các phép tính toán

7.2. Tăng, giảm giá trị thêm 1 đơn vị

`i = 5; k = i++;`

tương đương:

```
i=5;  
k=i;  
i=i+1;
```

`i = 5; k = ++i;`

tương đương:

```
i=5;  
i=i+1;  
k=i;
```

`i = 5; k = i--;`

tương đương:

```
i=5;  
k=i;  
i=i-1;
```

`i = 5; k = --i;`

tương đương:

```
i=5;  
i=i-1;  
k=i;
```


Chương 2: LẬP TRÌNH C TRÊN ARDUINO IDE

7. Các phép tính toán

7.3. Toán tử gán

Toán tử	Ví dụ	Tương đương với
=	<code>x = 5</code>	Gán 5 cho x
+=	<code>x += 5</code>	<code>x = x + 5</code>
-=	<code>x -= 5</code>	<code>x = x - 5</code>
*=	<code>x *= 5</code>	<code>x = x * 5</code>
/=	<code>x /= 5</code>	<code>x = x / 5</code>
%=	<code>x %= 5</code>	<code>x = x % 5</code>

Chương 2: LẬP TRÌNH C TRÊN ARDUINO IDE

7. Các phép tính toán

7.4. Trị tuyệt đối

abs() trả về giá trị tuyệt đối của một số

7.5. So sánh

constrain(x, a, b) Trả về giá trị trong khoảng giá trị cho trước [a, b]

Nếu $x < a$ → trả về giá trị **a**

Nếu $a < x < b$ → trả về giá trị **x**

Nếu $x > b$ → trả về giá trị **b**

Chương 2: LẬP TRÌNH C TRÊN ARDUINO IDE

7. Các phép tính toán

7.5. So sánh

min(x, a) Trả về giá trị nhỏ nhất trong 2 biến

Nếu $x < a$ → trả về giá trị **x**

Nếu $x > a$ → trả về giá trị **a**

max(x, b) Trả về giá trị lớn nhất trong 2 biến

Nếu $x < b$ → trả về giá trị **b**

Nếu $x > b$ → trả về giá trị **x**

```
int v1 = sqrt(9);  
int v1 = sqrt(9);
```

Chương 2: LẬP TRÌNH C TRÊN ARDUINO IDE

7. Các phép tính toán

7.6. Lũy thừa

```
int luythua1 = pow(2,3);
```

→ $\text{luythua1} = 2^3$;

7.7. Khai căn bậc 2

```
int v1 = sqrt(9);
```

→ $\text{v1} = 3$;

7.8. Bình phương

```
int v2 = sq(9);
```

→ $\text{v2} = 81$;

Chương 2: LẬP TRÌNH C TRÊN ARDUINO IDE

7. Các phép tính toán

7.9. e lũy thừa

$$y = \exp(x)$$

$$\rightarrow y = e^x;$$

Trong đó, $e = 2,71828\ 18284\ 59045\ 23536\dots$

7.10. Logarit tự nhiên

$$y = \log(x);$$

$$\rightarrow y = \ln(x);$$

7.11. Logarit cơ số 10

$$y = \log_{10}(x);$$

$$\rightarrow y = \log_{10}(x);$$

Chương 2: LẬP TRÌNH C TRÊN ARDUINO IDE

7. Các phép tính toán

7.12. Giá trị nguyên lớn nhất nhỏ hơn hoặc bằng

b = floor(a);

a = 1.2 → b = 1;

a = 1.9 → b = 1;

7.13. Giá trị nguyên nhỏ nhất lớn hơn hoặc bằng

b = ceil(a);

a = 1.2 → b = 2;

a = 1.9 → b = 2;

7.14. Chuyển đổi kí tự dạng chữ số về chữ số

const char Number[]="12345"
S = atoi(Number);

→ S=12345

Chương 2: LẬP TRÌNH C TRÊN ARDUINO IDE

7. Các phép tính toán

7.15. Random

Trả về một giá trị nguyên ngẫu nhiên
trong khoảng giá trị cho trước

random(min,max+1);

→ min và max là giá trị đầu và cuối
của khoảng giá trị mà hàm **random()** trả về

random(max+1);

→ giá trị min được hiểu ngầm bằng 0

* Để tạo ra dãy số ngẫu nhiên thật sự, dùng hàm **randomSeed(n)** trong hàm **setup()** với n là giá trị ngẫu nhiên bất kỳ

Chương 2: LẬP TRÌNH C TRÊN ARDUINO IDE

8. Xử lý chuỗi ký tự

8.1. Hàm `isAscii()` Kiểm tra kí tự đã truyền có thuộc bảng ASCII hay không

`bool isAscii(int c)`

8.2. Hàm `isWhitespace ()`

Kiểm tra xem kí tự đã truyền có phải là kí tự cách (' ') hoặc ('\t') không

`bool isWhitespace(int c)`

Chương 2: LẬP TRÌNH C TRÊN ARDUINO IDE

8. Xử lý chuỗi ký tự

8.3. Hàm isAlpha() Kiểm tra xem ký tự đã truyền có phải là chữ cái không

`bool isAlpha(int c)`

8.4. Hàm isalpha() Kiểm tra xem ký tự đã truyền có phải là chữ cái không

`int isalpha(int c)`

Chương 2: LẬP TRÌNH C TRÊN ARDUINO IDE

8. Xử lý chuỗi ký tự

8.5. Hàm `isAlphaNumeric ()`

Kiểm tra xem ký tự đã truyền có phải là chữ số không

`bool isAlphaNumeric (int c)`

8.6. Hàm `isalnum ()` Kiểm tra xem ký tự đã truyền có phải là chữ số không

`int isalnum(int c)`

Chương 2: LẬP TRÌNH C TRÊN ARDUINO IDE

8. Xử lý chuỗi ký tự

8.7. Hàm `isControl()`

Kiểm tra xem ký tự đã truyền có phải là ký tự điều khiển không

`bool isControl(int c)`

8.8. Hàm `iscntrl()`

Kiểm tra xem ký tự đã truyền có phải là ký tự điều khiển không

`int iscntrl(int c)`

BÀI TẬP CHƯƠNG 2

1. Giải thích hoạt động của đoạn chương trình sau:

```
const int ledPin = 13;
```

```
void setup()  
{  
  pinMode(ledPin, OUTPUT);  
}
```

```
void loop()  
{  
  digitalWrite(ledPin, HIGH);  
  delay(2000);  
  digitalWrite(ledPin, LOW);  
  delay(2000);  
}
```



BÀI TẬP CHƯƠNG 2

2. Giải thích hoạt động của đoạn chương trình sau:

```
void setup()
{
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT)
}
void loop()
{
  digitalWrite(2, HIGH);
  delay(500);
  digitalWrite(2, LOW);
  digitalWrite(3, HIGH);
  delay(500);
  digitalWrite(3, LOW);
  digitalWrite(4, HIGH);
  delay(500);
```

```
  digitalWrite(4, LOW);
  digitalWrite(5, HIGH);
  delay(500);
  digitalWrite(5, LOW);
  digitalWrite(6, HIGH);
  delay(500);
  digitalWrite(6, LOW);
  digitalWrite(5, HIGH);
  delay(500);
  digitalWrite(5, LOW);
  digitalWrite(4, HIGH);
  delay(500);
  digitalWrite(4, LOW);
  digitalWrite(3, HIGH);
  delay(500);
  digitalWrite(3, LOW);
}
```



BÀI TẬP CHƯƠNG 2

3. Giải thích hoạt động của đoạn chương trình sau:

```
#define LED 12
#define BUTTON 7
void setup()
{
  pinMode(LED, OUTPUT);
  pinMode(BUTTON, INPUT);
}
void loop()
{
  if ( digitalRead(BUTTON) == HIGH )
  {
    digitalWrite(LED, HIGH);
    delay(500);
    digitalWrite(LED, LOW);
  }
}
```



BÀI TẬP CHƯƠNG 2

4. Viết lại 8 bài tập chương 1 bằng cách sử dụng hàm

1. Sáng đuổi từ trái qua phải
2. Sáng đuổi từ phải qua trái
3. Sáng dần từ trái qua phải
4. Sáng dần từ phải qua trái
5. Sáng dồn từ trái qua phải
6. Sáng dồn từ phải qua trái
7. Chớp tắt xen kẽ
8. Chớp tắt xen kẽ từng cặp LED