

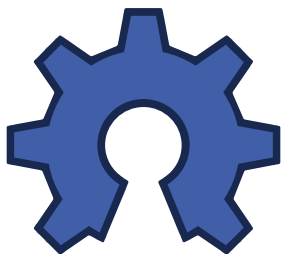
# Thiết kế trò chơi Trí Uẩn tương tác dựa trên bộ phần cứng mã nguồn mở Tự Học

Nhóm Đà Nẵng:

Hồ Tuấn Kiệt <[tuankiet65@gmail.com](mailto:tuankiet65@gmail.com)>

Nguyễn Hà Sinh <[s2tklnk@gmail.com](mailto:s2tklnk@gmail.com)>

Trương Thị Như Ý



open hardware



# Lời mở đầu:

---

Đây là bản thuyết minh cho sản phẩm phục vụ cho cuộc thi Tin học trẻ Toàn quốc năm 2014, bảng E2 (lập trình phần cứng cho khối Trung học Cơ sở).

Bản thuyết minh sẽ được chia thành các phần sau:

- Phần giới thiệu chung (trang 3):
  - Về các thành viên trong nhóm
  - Về sản phẩm
- Phần chính:
  - Giới thiệu về phần cứng lập trình (bộ phần cứng mã nguồn mở Tự Học) (trang 4+5)
  - Ý tưởng (trang 6)
  - Giao diện tương tác với người dùng (trang 7+8)
  - Các thuật toán có liên quan (trang 9)
- Phần phụ lục:
  - Các hình hỗ trợ và chú thích nguồn tham khảo(trang 10)

Tất cả mọi phần trong văn bản đều được phát hành dưới giấy phép [Creative Commons Attribution-ShareAlike 4.0 International](#), trừ những phần đã ghi chú ở trang 12. Mọi người có quyền chia sẻ hay chỉnh sửa bất kì phần nào của văn bản này dưới những điều kiện sau:

- Phải ghi nguồn tác giả và đánh dấu những phần đã chỉnh sửa từ văn bản gốc nếu có
- Nếu có chỉnh sửa thì phiên bản sau khi chỉnh sửa phải được phát hành với cùng giấy phép.

Các đoạn code thuộc **tài sản cộng đồng**, tức là ai cũng có thể dùng và sửa chữa mà không có một giới hạn nào. Nhưng hãy ghi nguồn nếu có thể.

Nhóm đã cố gắng để thông tin của bản thuyết minh được sát nhất với thực tế, nhưng có thể một số thông tin sẽ không còn chính xác do việc liên tục cải tiến từ đây đến ngày thi chính thức. Các thông tin chính xác nhất sẽ nằm trong bản thuyết trình vào ngày thi toàn quốc chính thức.

# Giới thiệu: Các thành viên trong nhóm:

---

## 1. Hồ Tuấn Kiệt (trưởng nhóm)

- Ngày sinh: 24/03/2000
- Trường: THCS Nguyễn Khuyến, Đà Nẵng
- Email: [tuankiet65@gmail.com](mailto:tuankiet65@gmail.com)

## 2. Nguyễn Hà Sinh

- Ngày sinh: 05/04/1999
- Trường: THCS Nguyễn Khuyến, Đà Nẵng
- Email: [s2tklnk@gmail.com](mailto:s2tklnk@gmail.com)

## 3. Trương Thị Như Ý

- Ngày sinh: 11/08/1999
- Trường: THCS Trưng Vương, Đà Nẵng

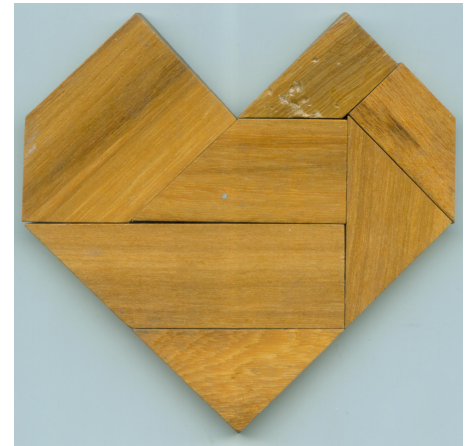
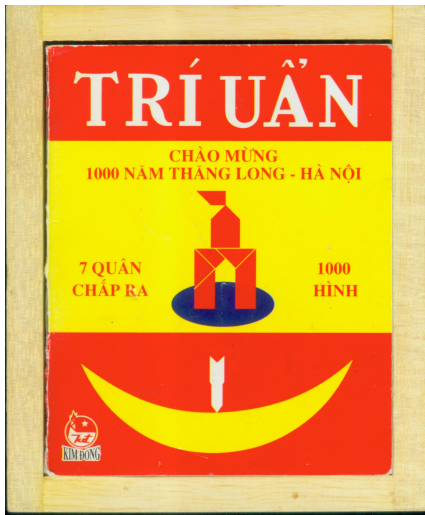
# Giới thiệu: Sản phẩm

---

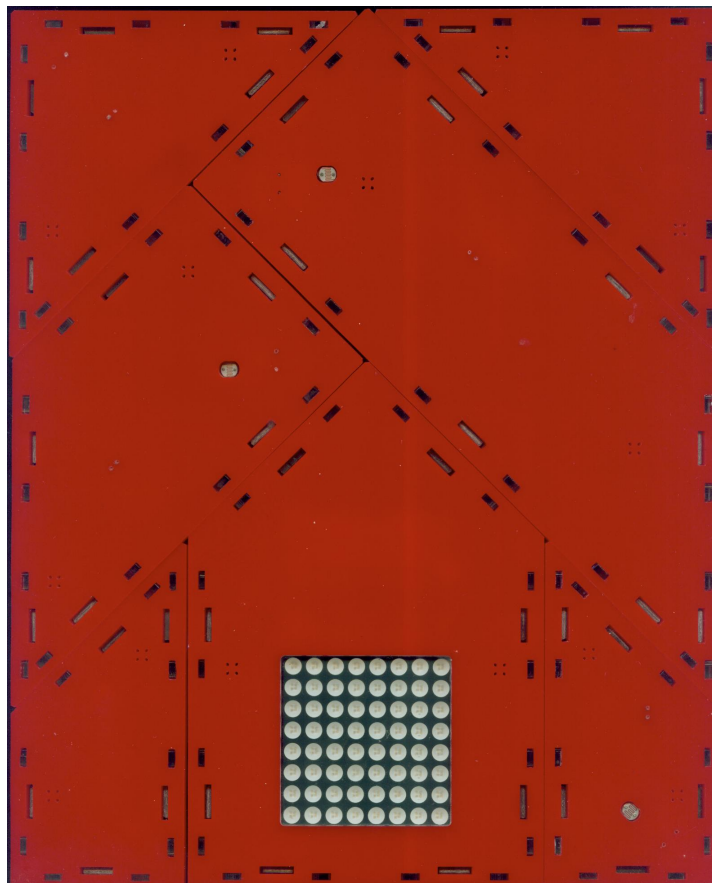
- Tên sản phẩm: **Trò chơi tương tác Trí Uẩn.**
- Dựa trên nền tảng phần cứng mã nguồn mở **Tự Học** của [Công ty cổ phần công nghệ DTT](#).
- Ngôn ngữ lập trình: **C/C++**
- Sử dụng các thư viện mã nguồn mở sau đây:
  - [thVLC](#) (Copyright 2014 quang.dinh@dttd.vn, GNU LGPL)
  - [thLedMatrix](#) (Copyright 2014 quang.dinh@dttd.vn, GNU LGPL)
  - [thAVR](#) (Copyright 2014 quang.dinh@dttd.vn, GNU LGPL)
  - [digitalWriteFast](#) (GNU GPL v3)
  - [IRremote](#) (Copyright 2009 Ken Shirriff and others)
  - Các thư viện **Serial**, **EEPROM**, **pgmspace** và các thư viện khác có sẵn trong Arduino IDE.
- Ngoài ra nhóm còn viết thêm 2 thư viện nữa là **thBuzzer** và **thIR** để đơn giản hóa việc sử dụng loa và điều khiển từ xa, và thư viện **message** phục vụ cho việc truyền tin.

# 1. Giới thiệu phần cứng lập trình

Trí Uẩn là một bộ trò chơi gồm 7 quân với hình dạng nhất định. Dựa theo lời của tác giả thì chỉ với 7 quân này có thể “xếp ra 1000 hình”



Bộ phần cứng mã nguồn mở Tự Học là sản phẩm nghiên cứu của Công ty cổ phần công nghệ DTT, dựa trên nền tảng phần cứng mã nguồn mở [Arduino](#). Bộ phần cứng bao gồm 7 khối khác nhau có hình dạng giống y hệt 7 quân trong trò chơi Trí Uẩn.



Khối trung tâm có một ma trận LED 8x8 có thể hiển thị màu xanh da trời, màu đỏ hay màu cam (pha giữa màu xanh da trời và màu đỏ). Sử dụng thư viện **thLedMatrix** để điều khiển ma trận LED. Có một buzzer (loa nhỏ) có thể sử dụng để báo hiệu. Loa sẽ kêu khi được cấp điện và chỉ có một mức tần số duy nhất là ~2700-2800Hz (không có tài liệu nên không có con số chính xác). Ngoài ra còn có thêm một cảm biến hồng ngoại, kết hợp với điều khiển từ xa đi kèm để điều khiển hoạt động của khối trung tâm. Dùng thư viện **IRremote** để nhận tín hiệu của điều khiển từ xa.

Khối trung tâm dùng vi xử lý **Atmel ATmega328P**, gồm 32kB flash (bộ nhớ để lưu chương trình), 2kB SRAM (Static RAM) và 1kB EEPROM (Electrically Erasable Programmable Read-Only Memory). Các khối khác dùng vi xử lý **Atmel ATmega8**, gồm 8kB flash, 1kB SRAM và 512B EEPROM.

Ở mọi khối đều có các cổng giao tiếp bằng ánh sáng nhìn thấy được (Visible Light Communication). Ánh sáng được truyền đi nhờ các đèn LED xanh ở các cổng. Ngoài ra ở các cổng còn có các điện trở quang để nhận tín hiệu. Ngoại trừ khối tam giác (2 khối ở 2 góc trên cùng như hình trên), các khối còn lại đều có thêm một điện trở quang ở một mặt để xác định tư thế của các khối (nằm úp hay ngửa). Sử dụng thư viện **thVLC** để gửi/nhận/xác định tư thế của các khối.

## 2: Ý tưởng

---

Hầu hết các nhóm khác đều chọn hướng đi là “tạo ra một bộ trò chơi mà khi người dùng ghép hình gì thì máy tự nhận ra hình đó. Nhưng nhóm nghĩ rằng hướng đi này có nhiều nhược điểm:

- Thời gian phát hiện lâu
- Không có/ít sự tương tác giữa người dùng và bộ trò chơi

Thay vào đó, nhóm chọn hướng đi khác, một bộ trò chơi mà người dùng sẽ chọn số của hình muốn ghép, sau đó người dùng ghép và bộ trò chơi sẽ kiểm tra xem người dùng có ghép đúng hay không.

Hướng đi này có các lợi ích:

- Trò chơi có tính tương tác với người dùng, người dùng hoàn toàn biết mình đang làm gì.
- Bộ trò chơi sẽ là một bộ trò chơi hoàn chỉnh hoàn toàn có tính thực tiễn cao.

Đến thời điểm viết thuyết trình, sản phẩm có thể làm được các việc sau:

- Cung cấp một giao diện tương tác thân thiện với người dùng
- Kiểm tra sự chính xác của hình ghép (với một hình ghép đúng thì thời gian dao động từ 20-30 giây)
- Có hệ thống tính điểm để ghi nhận thành tích của người dùng.

Dự kiến công việc còn lại đến ngày thi:

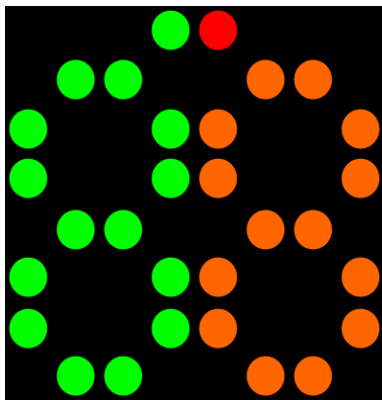
- Bổ sung các hình còn thiếu.
- Sửa các lỗi còn tồn đọng.
- Thêm một số hiệu ứng về đèn và một số tính năng nhỏ.

### 3: Giao diện tương tác người dùng

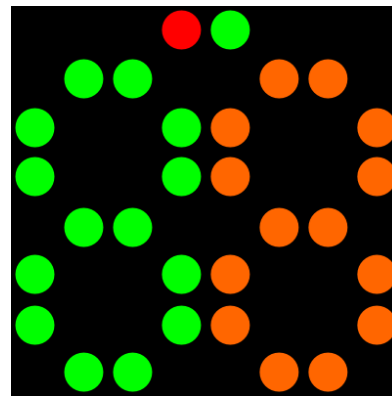
Đầu tiên, khi bật máy, người dùng sẽ ở màn hình thứ 0, tức là màn hình khởi động. Loa sẽ kêu trong 1000ms và hiệu ứng đèn LED khởi động bắt đầu.

Tiếp theo, người dùng sẽ đến màn hình thứ 1, gọi là màn hình chọn số và hiển thị số điểm số. Gọi như thế vì ở khi ở màn hình thứ 1, người dùng có thể nhấn nút CH-/CH/CH+ trên điều khiển từ xa để thay đổi giữa việc chọn màn hình chọn số (ta gọi là màn hình 1.1) và màn hình hiển thị số điểm (ta gọi là màn hình 1.2). Người dùng có thể biết được đang ở màn hình 1.1 hay 1.2 bằng cách xem màu của pixel ở tọa độ (4, 0). Nếu pixel ở tọa độ (4, 0) có màu xanh, tức là người dùng đang ở màn hình 1.1, ngược lại nếu có màu đỏ, người dùng đang ở màn hình 1.2.

Ở màn hình 1.1, giới hạn của số là từ 0 tới 99 (số 0 là số đầu tiên, tức là có thể lưu trữ tối đa 100 hình). Ở màn hình 1.2, giới hạn của số là từ 0 tới 100 (0 tức là người dùng chưa giải được hình nào, 100 tức là người dùng đã giải hết tất cả các hình có thể).



Màn hình 1.1



Màn hình 1.2

Giả sử ta đang ở màn hình 1.1. Ta gọi  $N$  là số hiện trên màn hình (khi bật lên thì  $N$  ban đầu bằng 0). Lúc này người dùng có thể sử dụng các nút trên điều khiển từ xa để thay đổi số như sau:

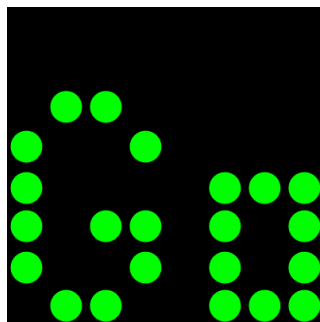
- Nút NEXT/VOL+: Tăng  $N$  lên 1 đơn vị.
- Nút PREV/VOL-: Giảm  $N$  1 đơn vị.
- Nút FOL-/100+: Tăng  $N$  lên 10 đơn vị.
- Nút FOL+/200+: Tăng  $N$  lên 20 đơn vị.
- Nút EQ:  $N = N / 10$
- Nút số từ 0..9: gọi  $X$  là nút số đã nhấn:
  - Nếu  $N < 10$  thì  $N = N * 10 + X$
  - Nếu  $N > 10$  thì  $N = (N \% 10) * 10 + X$

Với các nút NEXT/VOL+/FOL-/FOL+/100+/200+, nếu như  $N$  sau khi tăng lớn hơn 100 thì  $N$  sẽ bị giảm đi 100 đơn vị. Với các nút PREV/VOL-, nếu như  $N$  sau khi giảm bằng -1 thì  $N$  được gán bằng 99.

Ở màn hình 1.2, người dùng không thể dùng các nút như trên để thay đổi số điểm.

Sau khi chọn số xong ở màn hình 1.1, người dùng sẽ nhấn nút PLAY/PAUSE và được đưa đến màn hình 2.

Ở màn hình 2, ma trận LED sẽ hiển thị chữ "Go" 3 lần, mỗi lần nháy cách nhau 300ms. Sau đó màn hình sẽ hiển thị màu đen hoặc có thể có thêm hiệu ứng tự chọn. Màn hình 2 là lúc người dùng ghép hình theo đúng số hình đã chọn ở màn hình 1.1.



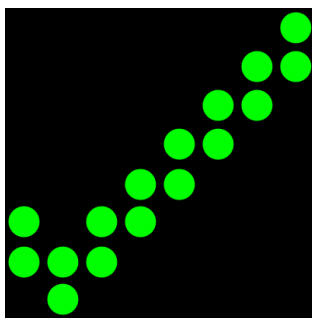
Màn hình "GO"

Sau khi ghép xong, người dùng nhấn nút PLAY/PAUSE và được đưa đến màn hình 3. Màn hình 3 có 3 dấu chấm trên ma trận LED, báo hiệu bộ trò chơi đang kiểm tra xem người dùng có ghép đúng hình hay không. *Thuật toán để kiểm tra các hình ghép không nằm trong khuôn khổ bài thuyết minh, tuy nhiên nó khá giống với các **thuật toán đệ quy**.*

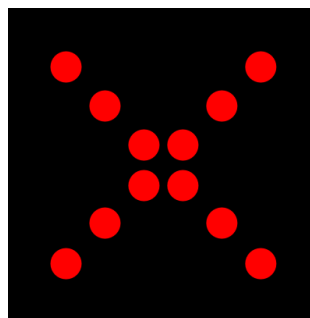
Sau khi kiểm tra xong, người dùng sẽ được đưa đến màn hình 4. Màn hình 4 hiển thị kết quả kiểm tra, gồm 2 phần là màn hình 4.1 và màn hình 4.2. Nếu người dùng ghép đúng hình đã chọn thì màn hình 4.1 sẽ được hiển thị, ngược lại thì màn hình 4.2 sẽ được hiển thị.

Màn hình 4.1 thể hiện một dấu tick ✓ có màu xanh, cho biết người dùng đã ghép đúng. Sau khi hiển thị xong thì bộ trò chơi sẽ kiểm tra xem người dùng đã giải được hình này trước đây hay chưa, nếu chưa thì bộ trò chơi tự động đánh dấu hình đó là đã giải được và tăng số điểm lên 1. *Thuật toán xác định xem hình đã giải từ trước đó chưa sẽ được đề cập ở trang 9.*

Màn hình 4.2 thể hiện một dấu chéo ✗, cho biết người dùng đã ghép sai. Ở cả 2 màn hình thì dấu tick/chéo đều nhấp nháy, mỗi lần nhấp nháy cách nhau 500ms đối với dấu tick và 250ms đối với dấu chéo.



Màn hình 4.1



Màn hình 4.2

Sau khi hiển thị xong màn hình 4.1/4.2 thì người dùng sẽ được đưa về lại màn hình 1.1 với số của hình đã lựa trước đó. Vòng lặp này sẽ lặp lại mãi cho đến khi người dùng tắt máy.



# Thuật toán: Nhận biết hình đã giải trước đó chưa

Một chút về EEPROM. EEPROM (Electrically Erasable Programmable Read-Only Memory) là một loại bộ nhớ thuộc dạng non-volatile (tức là dữ liệu không mất đi khi mất nguồn, trái ngược với SRAM). Ở bộ vi xử lý Atmel ATmega328P, EEPROM có dung lượng 1kB. Ở bộ vi xử lý Atmel ATmega8, EEPROM có dung lượng 512B.

Ta gọi số của hình cần kiểm tra là  $N$ .

Thường ta sẽ giải quyết vấn đề này một cách đơn giản bằng cách kiểm tra xem ô nhớ ở địa chỉ EEPROM  $N$  có giá trị là 0 hay 1. Nếu là 0 thì hình chưa giải, và ngược lại. Nhưng cách này chiếm dụng nhiều bộ nhớ.

Do vậy để sử dụng bộ nhớ một cách tối ưu hơn, ta sẽ dùng **bit**.

Đầu tiên, ta xác định xem địa chỉ EEPROM thứ mấy sẽ chứa biến biểu thị tình trạng của hình ghép thứ  $N$  bằng phép tính  $M=N/8$

Sau đó, ta đọc từ EEPROM dữ liệu ở ô nhớ thứ  $M$ , và lưu ở một biến tạm gọi là  $P$ .

Ta thực hiện phép toán bit  $Q=P \& (1 << (N \% 8))$  nhằm kiểm tra tình trạng của bit chứa trạng thái của hình  $N$ . Nếu  $Q=0$  thì hình thứ  $N$  chưa được giải, và nếu ngược lại ( $N \neq 0$ ) thì hình thứ  $N$  đã được giải.

Trong trường hợp hình  $N$  chưa được giải, ta đổi trạng thái của bit đó bằng phép toán  $R=P | (1 << (N \% 8))$  và gán giá trị của ô nhớ ở địa chỉ EEPROM  $M$  bằng  $R$ .

Giải pháp sử dụng bit chỉ sử dụng 13 ô nhớ trong EEPROM, trong khi đó giải pháp đơn giản ban đầu sẽ cần phải sử dụng 100 ô nhớ.

Vì EEPROM có số lần đọc/viết giới hạn (~100.000 lần) do đó phần mềm sẽ cache trước các số vào SRAM.

Mã nguồn tham khảo:

```
unsigned char bitmapCache[13]={0};
void scoreUpdate(unsigned char score, unsigned char shapeNum) {
    unsigned char a=shapeNum/8;
    if (!(bitmapCache[a]&(1<<(shapeNum%8)))) {
        bitmapCache[a]|=(1<<(shapeNum%8));
        EEPROM.write(100, score+1);
        EEPROM.write(a, bitmapCache[a]);
    }
}
```

## Phụ lục: Các hình đã ghép được

---

Về mặt lý thuyết, số lượng các hình mà bộ trò chơi có thể ghép được phụ thuộc vào dung lượng flash còn lại trên khối chính. Trên thực tế, máy hỗ trợ tối đa 100 hình vì giới hạn chữ số. Từ đây đến ngày thi, nhóm sẽ cố gắng bổ sung thêm hình. Danh sách dưới đây ghi các hình mà máy hỗ trợ cho đến thời điểm viết bản thuyết minh này.

Tên hình (trang X, hình Y) có nghĩa là hình có tên là *Tên hình*, nằm ở trang X trong sách Trí Uẩn đi kèm và có số hình là Y trong sách Trí Uẩn. Dấu ?? biểu thị việc hình không nằm trong sách Trí Uẩn.

Các hình đều có hình ảnh đi kèm, có thể tải tại [đây](#).

0. Bảng đen (trang 11, hình 20)
1. Búa (trang 19, hình 53)
2. Cây (trang 35, hình 105)
3. Chữ thập (trang ??, hình ??)
4. Chuông chùa (trang 31, hình 91)
5. Chim trời (trang 16, hình 41)
6. Con ốc sên (trang ??, hình ??)
7. Số một (trang 27, hình 79)
8. Hạt kim cương (trang 31, hình 93)
9. Trái tim (trang 36, hình 11)

## Phụ lục: Tham khảo

---

- **Open Hardware** logo: <http://oshwlogo.com/>
- Logo và hình ảnh của bộ phần cứng mã nguồn mở **Tự học**: Scan từ bản gốc. Thiết kế thuộc bản quyền của [Công ty Cổ phần Công nghệ DTT](#).
- Tài liệu của vi xử lý ATmega328p: [http://www.atmel.com/Images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P\\_datasheet\\_Summary.pdf](http://www.atmel.com/Images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Summary.pdf)
- Tài liệu của vi xử lý ATmega8: [http://www.atmel.com/Images/Atmel-2486-8-bit-AVR-microcontroller-ATmega8\\_L\\_summary.pdf](http://www.atmel.com/Images/Atmel-2486-8-bit-AVR-microcontroller-ATmega8_L_summary.pdf)
- Hình của màn hình LED được tạo bằng **Inkscape 0.48.5**
- Code beautified bằng [Code Colorizer Formatter](#)
- Hình ảnh bộ trò chơi Trí Uẩn: scan từ bộ gốc. Bản quyền thuộc về [Công ty TNHH Trí Nguyễn](#).