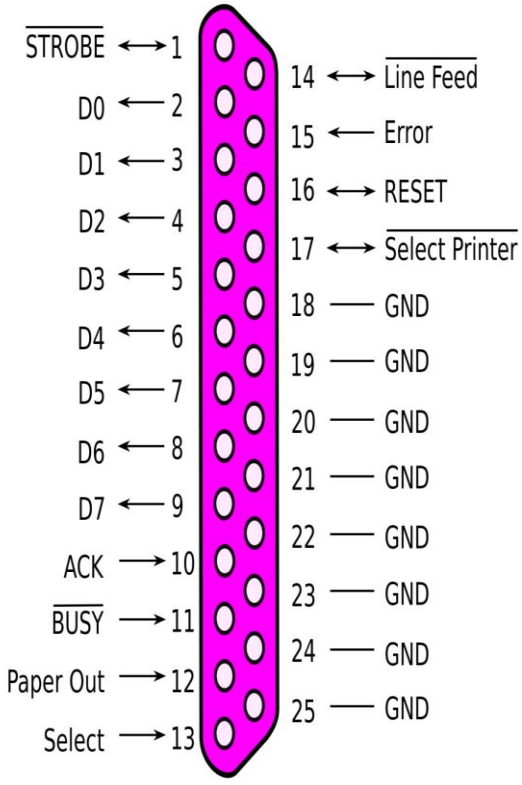
ROBOT ARM PROJECT

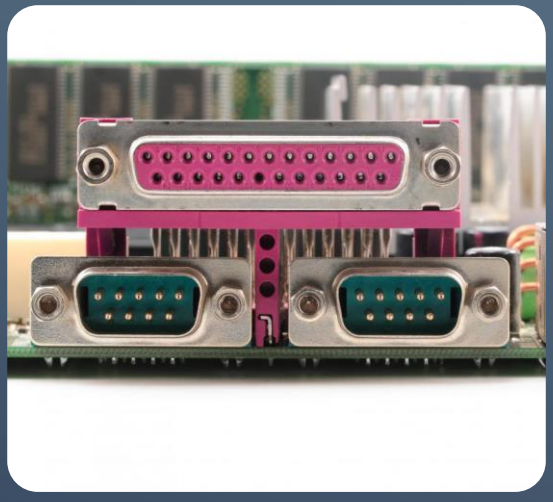
Using LPT (Line Printer Parallel port)

- مرتضى حكمان
- حسين حمزة
- احمد حازم
- ياسر صلاح
شعبة A

WHAT IS LINE PRINTER PORT(LPT)?

The Parallel Line Printer Port (LPT) is an old interface on computers used to connect peripherals, especially printers. It transfers multiple bits of data at once, which is why it's called "parallel". The parallel port has exactly 8-bits of data are used to output ASCII code to the printer
In our project we're going to use this byte or these 8 bits to control a robot arm controlled mainly by stepper motors and the parallel port



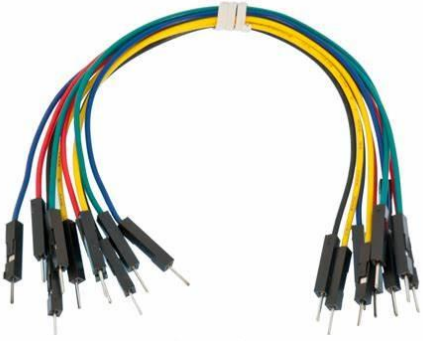


HARDWARE SPECS:

3 STEPPER MOTORS

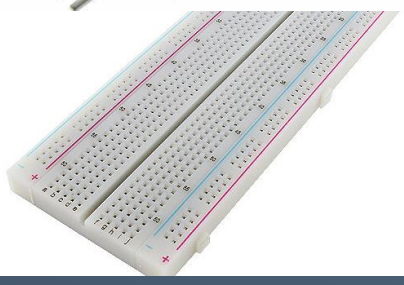
3 STEPPER CONTROLLORS

JUMPER WIRES



3 NPN TRANSISTORS

BREAD BOARD



THE MOST IMPORTANT THING:

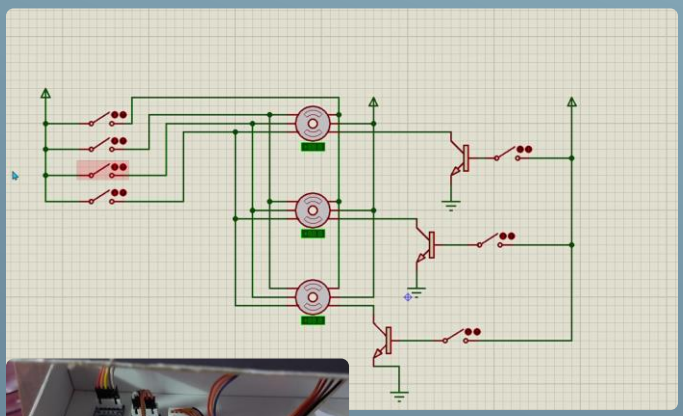
GETTING AN OLD PC !

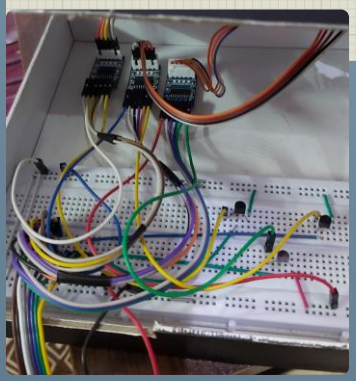
THE MOST CHALLENGING PART OF THIS PROJECT IS GETTING A PC WITH THIS PORT, AND NOT JUST A PC, BUT A WORKING ONE !, WE SEARCHED A LOT TRYING TO GET A DESKTOP THAT MAY HAVE THIS PORT, BUT THEY EITHER DEAD OR HAS A LOT OF UNSOLVABLE ISSUES
UNTILL WE FINALLY GOT ONE, FOR JUST 20$.
BUT THIS WASN'T THE END OF THE STORY WE SPENT 5 DAYS CONTINOUSLY FIXING ISSUES OF THIS PC, LET ME JUST LIST SOME OF THESE ISSUES :
- NOT WORKING MOUSE AND KEYBOARD (CORRUPTED SYSTEM THAT ONLY SUPPORTS PS-2 MOUSE AND KEYBOARD
- CAN'T INSTALL NEW WINDOWS (DOESN'T SUPORT FLASH BOOT, DEPRECIATED STORAGE DEVICE)
- PARALLEL PORT DOESN'T WORK UNTIL YOU DO RESTART
…., AND THE LIST GOES ON
BUT AT THE END WE MANAGED TO FIX ALL THESE ISSUES BY RESEARCHING AND TRYING, UNTILL WE WERE ABLE TO COMPLETE THIS PROJECT

IMPLEMNTATION

CHOOSING THE STEPPER MOTOR:

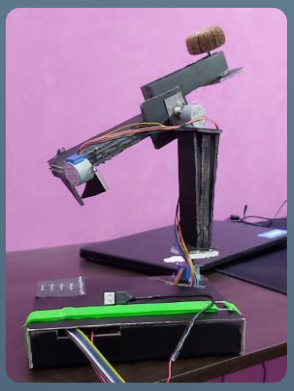
FOR CHOOSING EACH STEPPER MOTOR, WE CONNECTED 3 TRANSISTOR NPN TYPE THAT WHEN IT GETS A 1 BIT SIGNAL IT ACTIVATE THE WANTED STEPPER MOTOR, LOWERING THE NUMBER OF BITS USED IN THE PARALLEL PORT

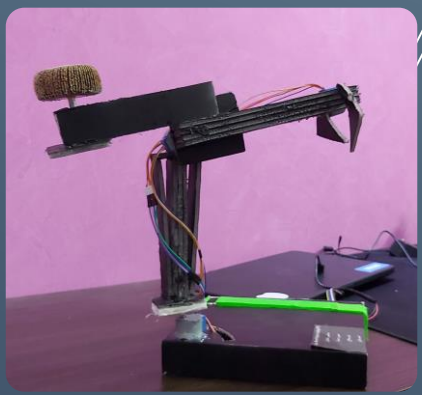


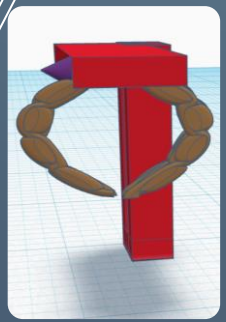


MOVING THE STEPPER MOTOR:

EACH STEPPER MOVES ETHER CLOCK WISE OR COUNTER CLOCK WISE TO (ROTATE,MOVE-UP-DOWN,PICK-UNPICK) WHICH IS CONTROLLED BY THE KEYBOARD ATTACHED TO THE PC AND DURING THE SOFTWARE RUINING.







CHECK VIDEO PROVIDED IN FILES TO DEMONSTRATE THE WORKING PROJECT

SOFTWARE

A LOOK AT SOME CODE SNIPPETS

DEFINING PORTLOCATION, STEPPER STEPS, TRANSISTOR VALUES

```
File  Edit  Search  Run  Compile  Debug  Project  Options  Window  Help
                    NONAME00.CPP                                  1
[■]                 NONAME01.CPP                                  2=[↑]

#define PortAddress 0x378
#define TR1 0x80
#define TR2 0x40
#define TR3 0x20
int cw[7] = { 0x01, 0x03, 0x02, 0x06, 0x04, 0x0c, 0x08 };
int ccw[7] = { 0x08, 0x0c, 0x04, 0x06, 0x02, 0x03, 0x01 };
int k = 0;
int pos = 0;
int true = 1;
char input;
char SelectMask;

14:1
```

CLOCK WISE FUNCTION

```
File  Edit  Search  Run  Compile  Debug  Project  Options  Window  Help
                    NONAME00.CPP                                  1
[■]                 NONAME01.CPP                                  2=[↑]
void CW(int rotationDegree) {

  k = pos;


  for (int i = 0; i < rotationDegree; i++) {
    if (k >= 7) { k = 0; }

    cout << "CW" << endl;
    outportb(PortAddress, cw[k]);
    delay(2);
    k++;
  }
28:1
F1 Help  F2 Save  F3 Open  Alt-F9 Compile  F9 Make  F10 Menu
```

PART OF MAIN FUNCTION

```
File  Edit  Search  Run  Compile  Debug  Project  Options  Window  Help
                    NONAME00.CPP                                  1
[■]                 NONAME01.CPP                                  2=[↑]
void main() {
  clrscr();

  while (true == 1) {
    for (int i = 0; i <= 6; i++) {
      cw[i] = cw[i] & 0x1f; //make last 3 bit zeros
      ccw[i] = ccw[i] & 0x1f;
    }
    cout << "enter number of stepper" << endl;
    SelectMask = getch();  //take char and jamp to another line
    if (SelectMask == '1') {
      for (int i = 0; i <= 6; i++) {
70:1
F1 Help  F2 Save  F3 Open  Alt-F9 Compile  F9 Make  F10 Menu
```

PLEASE CHECK PROVIDED CODE AND VIDEO DEMONSTRATION TO GET MORE IN DEPTH WITHT THE PROJECT

END