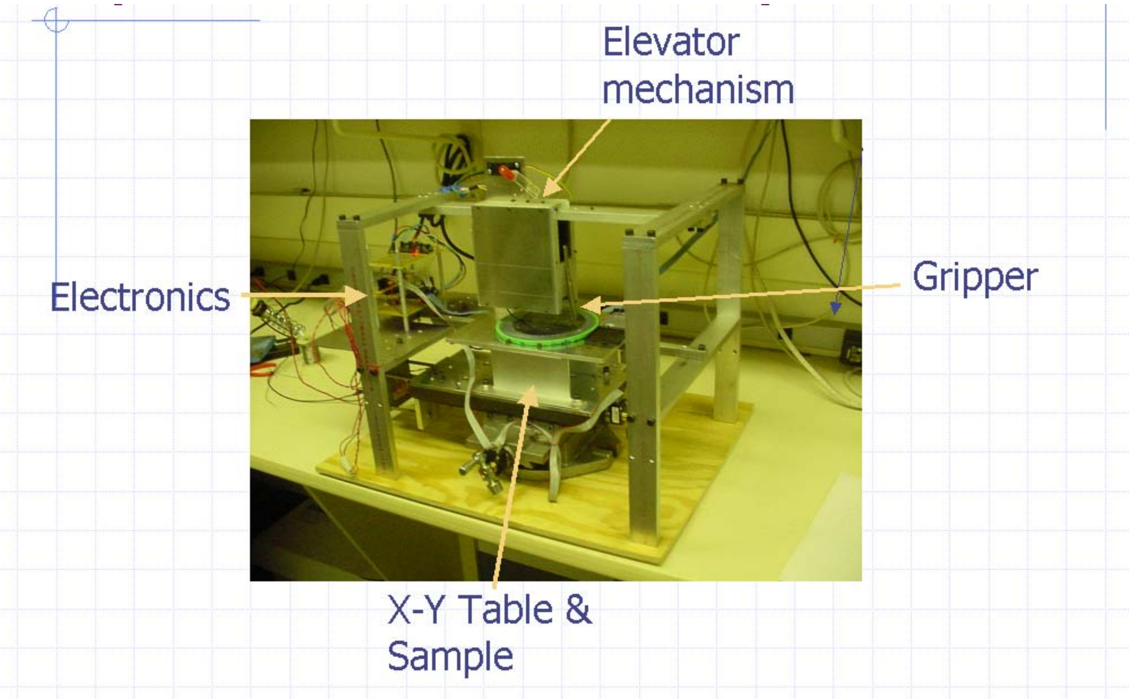
Introduction

The goal this year is to create a mechatronic device capable of picking and placing of silicon die. The die are extremely small and placing them into storage packs by hand would take a lot of time. This tedious manual task may be automated. There are restrictions, however, like not being able to touch the top of the die because the important MEMS parts are located there. By creating such a system, basic control, electronics, mechanisms and system integration are naturally taught.

Inspiration

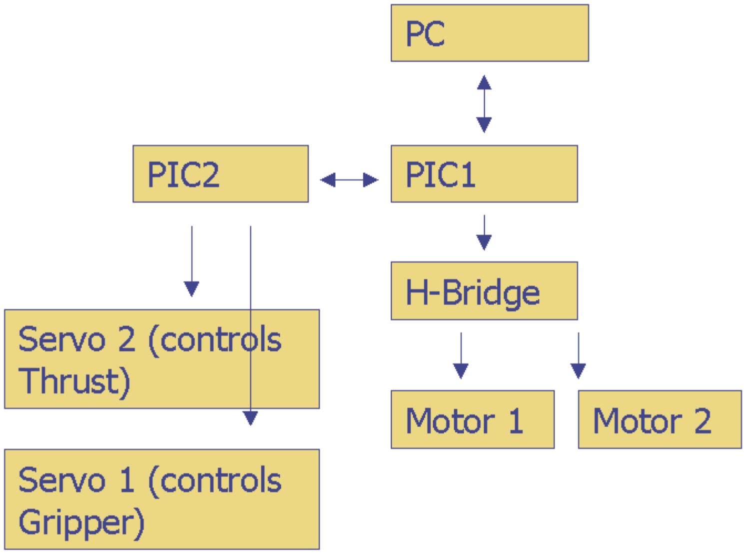
Our final system is comprised of four major components. These are the elevator mechanism which allows our gripper to move up and down, our gripper that opens and closes, our X-Y Table that moves our sample around and the electronics to control everything.



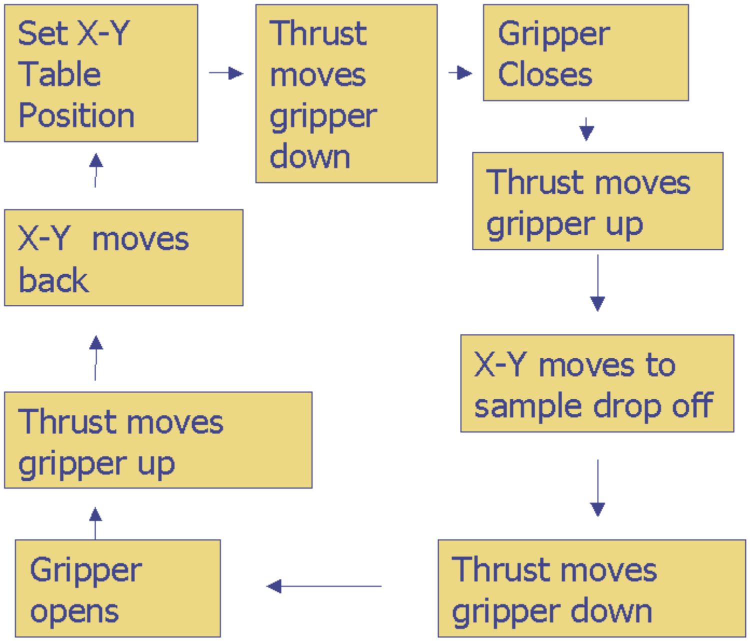
When we originally designed the system, we wanted to keep it as simple as possible. This meant the least number of parts had to be integrated and the least number of parts should be used. Initially with our first system design, we took our limited knowledge of mechatronic systems and designed what we believed would be a simple system. However, we soon found out that there were often easier methods to accomplish the tasks needed so many items were changed eventually. As time progressed, our system may have become even simpler. One of our initial ideas was to forego sensing for the time and base our location purely on initial manual alignment and then automation. We left in our schematics the possibility of the addition of sensing but we need not build sensing into our system in the end.

System

The physical components are indicated in our schematic below. The servos control the elevator shaft and the opening and closing of the gripper.



A system flowchart shows how the system operates.

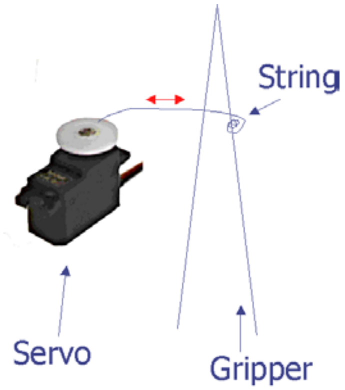


Our goal for this project was to be able to pick up one chip reliably first so our design revolved around that aspect. We learnt a lot from having to use two PICs to communicate even though at times it was frustrating.

Subsections

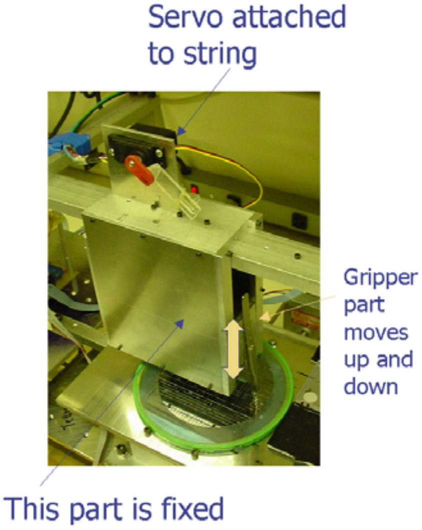
Overall Gripper:

Our initial idea was to use nitinol wire. We quickly found out that it did not compress to the lengths that we needed. We soon replaced it with a servo which was simple to control.



Overall Elevator.

Initially we planned on using a worm gear to drive the shaft up and down. However, this proved problematic and once again we decided to use a servo instead.



Overall Mechanical:

Supporting Board: to support the whole structure.
X-Y Table: to satisfy the wafer to move in x-y directions precisely.
Four Bars: to support the supporting beams and elevator on it.
Upper Supporting Beam: to support one motor and the elevator to satisfy the z-direction moving function.
Lower Supporting Beam: to support the screw to satisfy the function of punching the target chip up.
Elevator: to support the other motor and inside tweezer to satisfy the gripping function.
Electronics Supporting plate: to support the electronics and PIC boards.
Wafer Supporting plate: to mount the wafer and chip container.

Most of our components are made of steel because they satisfy the strength requirement of our design and they are easily acquired and machined. Besides, most of our components are machined from the pieces in ECE/ME stockroom. The materials in stock have standard dimensions. By doing this, we can save the time and money to machine the parts. The beams and supporting bars are designed with thicker shapes since the total frame is supposed to be very stable. Any bending or deflection of the frame will result in some uncertain error in assembling or operating.

The elevator mechanism is designed with smaller dimensions because originally the upper supporting beam was designed to be rotated and controlled by a larger rotation arm servo. The beam will bend and the performance will be low if the load is too heavy. A flat thin tipped tweezer is selected because of its excellent performance to pick up chips. For a non-flat or thicker tipped tweezer, the performance of operation will decrease. We designed three holes drilled on both legs of the tweezer; two are used for mounting and the third one is used to control the position of the tweezer. The servos are selected to offer the minimum torque/force at least! For saving space and money, we use servos as small as possible!

Gripper:

Our tweezer is made of very hard material and its structure is very tiny; therefore, it's impossible to drill some holes on it by ourselves. We also need to fix one leg of our tweezer on a plate to prevent it from vibrating when we tie up the rope to control the tweezer to pick up the chip. But by doing so, our tweezer will incline with a small angle. The side-effect results in a little difficulties to pick up a chip.

We still need to improve the control part to move our tweezer to open at appropriate position. For different chip size, we need to have the tweezer satisfy each requirement.

I think we still need to change our motor to a powerful one to increase the force to grabbing the chip since sometimes we can not pick up some chips successfully.

Elevator:

It's difficult to build an elevator so accurately for the tweezer to slide upward and downward smoothly. Originally I want to design some rollers on the elevator, but it seems to be time-consuming and not easy to do this. Since we consider the power of motor is very small, the thickness of the elevator is designed to be very thin and we can only do a little machining on it. Precise dimensions are very important in this part and we also use WD-40 to improve the friction obstacle.

Selection the rope to pull the elevator is another problem. First, we use a wire to serve but we failed finally since the friction produced is so large. We change the our design to nylon rope for its merits of high strength, light weight, cheap , and low friction coefficient.

We also need to improve the control part to move the elevator downward smoothly; otherwise, the tweezer will hit the wafer and result some damage on the chips.

Other mechanical challenges:

We need to set up all of each components precisely since position controlling is an important part in this project. We need to revise some of our parts if there is some machining error.

One way to prevent spending extra time to machining the components is to design it more flexiblely. To drill two or three more holes will make our design adjustable.
One of our major design mistake is to use a fixed screw to punch the target chip. It will restrict the motion range of x-y table in the small area. I think a controllable puncher will be a better design.

Electronics:

Our plan for our electronic systems from the start was for them to be as simple and robust as possible. Several of us had some experience with just how badly complex electronic systems could fail if things went wrong and didn't want to see that happen to this project. As such, our electronics was built around proven designs – those provided by the class.

Building those components came with its own share of problems. The actual assembly of our first PIC board was relatively uneventful, though the actual programming of it presented a few problems in syntax as our programmers were used to c++, not c naming conventions. This brought about a great deal of cursing and hair pulling until whatever error we were having presented a simple (and obvious when we looked back on it) answer.

Construction of the motor controller appeared to be simple, as it was just a little bit of solder work in a fairly large area. However, it was at this point that we started becoming acquainted with the demons that would haunt us throughout this project – connectors and slight errors caused by inattention to details, with its ancillary problem of lack of documentation and communication of the state of projects between team members. Connectors at first appeared a small problem, as the first demonstrations required by the class were simple and the rat's nest of wiring we created was fairly small; the occasional outages in control signals were easily overlooked. The inattention to detail manifested in misplacing two of the capacitors on the board, resulting in weird behaviors (most notably the slow speed of the motors we demonstrated). We noticed the second problem soon enough (within a week), but the first would plague our group through the rest of the project.

Our original design for the project was very vague, but basically called for the user to input commands from the computer, have those commands processed on board the PIC and then the appropriate motors moved with appropriate feedback to the user.

What this translated into with electronics was basically a serial port connector to the PIC, using the onboard rs232 to provide communications. From that point we took the pwm signal offered by the PIC and routed it to a motor controller, as well as the various enable lines to pick which motor moved where.. We at first thought that we would only need 1 PIC and 2 motor controller, but we changed this design later…more on that a bit further down.

(note: should include a flowchart/diagram from the presentation here probably)

Our first design checkpoint was to create a working gripper model. The initial tests of muscle wire showed that it would be unsuitable for the gripper design, but we used one of the principles of the wire – using wire tension to close a common tweezer – as the basis of our modified design, which utilized a servo in place of the muscle wire.

This was painless, as the mechanical construction was extremely simple and the electrical connections amounted to 3 wires, not counting power.

Our next checkpoint was to show a working x-y table movement program. Here we started running into serious problems. The first of these was the fact that our xy table was faulty in a couple of key ways which made it impossible to show one of the directions working at our demonstration, the second was that the rats nest of electrical connections (mostly from the PIC to the motors and encoders) was having a noticeable effect on performance…a bad one. Intermittent signals were causing bad encoder readings, though at this point we hadn't tried to make the connectors 'robust', so this didn't stop the table from functioning. More importantly, the system was basically hanging together on wire tension and prayer. If you breathed on it wrong things would stop working. After the demonstration, we resolved to improve the connections of the system and make it far more robust.

Here our real pain with connecters occurred. In our attempts to make the system more robust, we failed to use the wire crimps correctly. This didn't really cause a problem when we tested the wires independently – the crimps had fairly solid contacts. What really killed signals was the fact that when we placed the various wires into the black plastic sheaths to prevent short circuits, the connecters would basically stop functioning. This made debugging the improved user interfaces and added functionality we needed for the x-y table extremely hard.

On the positive side, our attempts to improve the power connections went very well, as these proved quite resistant to being moved around and generally mistreated; this no doubt saved at least one of us at some point from electrocution.

(if we have any images of the connecters we used for power, might want to place them here)

It was around this point that we also changed our designs for the arm mechanism and realized we needed 2 PIC boards and only 1 motor controller. Communications between the PIC's was to be handled through interrupts. Only one of the PIC's would be connected to a computer via a serial port, and as such modification of values on the second PIC through a user interface would be somewhat limited. This wasn't deemed to be a problem because of the simplicity of the tasks assigned to this PIC (this later proved a sound assumption, as this part of our design worked near perfectly).

We couldn't really build this part of our electronics until we had a solid working user interface/xy table movement program, and attempts to make connectors between the encoders and our first PIC showed serious problems. Eventually, we realized that this part of our project was a serious problem and we asked for advice in construction of our connectors. Utilizing this advice created a profound change in both the robustness and the functionality of our system. Debugging suddenly became a nearly pure software issue rather than jiggling wires around hoping that the connection would remain solid enough to get some useful data.

Final construction from this point went fairly smoothly. There were some small issues where the PIC's didn't seem to be communicating correctly, but this was resolved in software.

Resources

2 * SMT-4 Precision Tweezers. http://www.regine.ch/e_05.htm. $70.

1 * High Speed BB Micro Servo. http://www.acroname.com/robotics/parts/R28-MX-50HP.html. $23.50
2 * Futabo Servos. Provided by TAs.
ECE Machine Shop Work and Parts. Unknown cost.
Various Small Items from CMU Art Store (Glue, scissors, foamboard). $15.

Insights

Looking back at the progress our group made and the problems we encountered, it is safe to say we learned a few things.

First and most important, we learned that good communication channels between group members is *vital*. At first this didn't seem an issue, as our mechanical component wasn't required for several of the checkpoints we set. However, when it came time to integrate the various systems, we realized that certain things weren't finished or perhaps even started because of miscommunications.

Though the importance of the lab notebooks as tools for this sort of communication was stressed over and over, our group didn't really utilize this as there was no real overlap in our responsibilities. We relied more on email and discussion to set timetables, but this proved inadequate. Doing this again, I'm sure we would pay more attention to documentation and setting timetables that we stuck to rather than the more informal system we set up.

Next would be that simple designs are good. Though we based our entire project on this principle there is no reason not to reiterate the point. The fact that so many of our systems were so simply designed is what saved our group from not being able to finish the project when all sorts of problems started cropping up near the end of the class.

However, this was also a good illustrator of a problem we had – our designs were in limbo for far too long. Many of the problems we had near the end could have been ironed out earlier had the parts been available. We should have locked down what we were building several weeks before we really did (there could be an argument made that we didn't really lock down our design until our presentation, but the point I mean is a little after spring break). This ties into group communication and lack of timetables (or at least sticking to which ones we had).

As to the design itself, things we would probably change if given the chance would be to add a motor to drive a plunger upward into the tray rather than relying on a fixed point to provide enough of a curve in the silicon tray to allow our gripper to function. This would also alleviate the problem we had with this fixed point limiting where we could move the xy table. Adding a smoother downward motion for the elevator would be nice. A more automated retrieval system on the software side would also be on our wishlist, as would be an easier way to set the initial points for the dead reckoning software.