Electronic Design Lab Project
Barcode Scanner

INDIAN INSTITUTE OF TECHNOLOGY, BOMBAY

Project Code : BT06

Team Members:

Vrihad Agarwal	150100004
Shashwat Shukla	150260025
Arpit Singh	150070059

Faculty Mentor: Siddharth Tallur
TA/RA: Chandan Choudhary

Abstract

A Barcode is a pictorial method of representing a numerical code, using black and white vertical bars (filled rectangles). In most of the commonly used barcode encoding standards, a fixed number of bits (0 or 1) are used to represent a character, where each bit is represented by a bar of unit width, and the color of the bar is decided by the value of the bit. ZERO- bit corresponding to WHITE bars and ONE-bit corresponding to BLACK bars. These standards were developed to make it easy for machines to decode the barcodes.

A **barcode reader** (or **barcode scanner**) is an electronic device that can read barcodes printed on a surface, and display or send the decoded information to a computer.

The aim of this project is to develop a prototype of an **indigenous** barcode scanning engine to convert a 1D printed barcode to a fixed length stream of bits and decode it on a microcontroller to obtain the final decoded information. In this project, we have built such a barcode scanner and have scanned barcodes adhering to the Universal Product Code (UPC-E) standard. We then display the final decoded output on a laptop, transferred via USB.

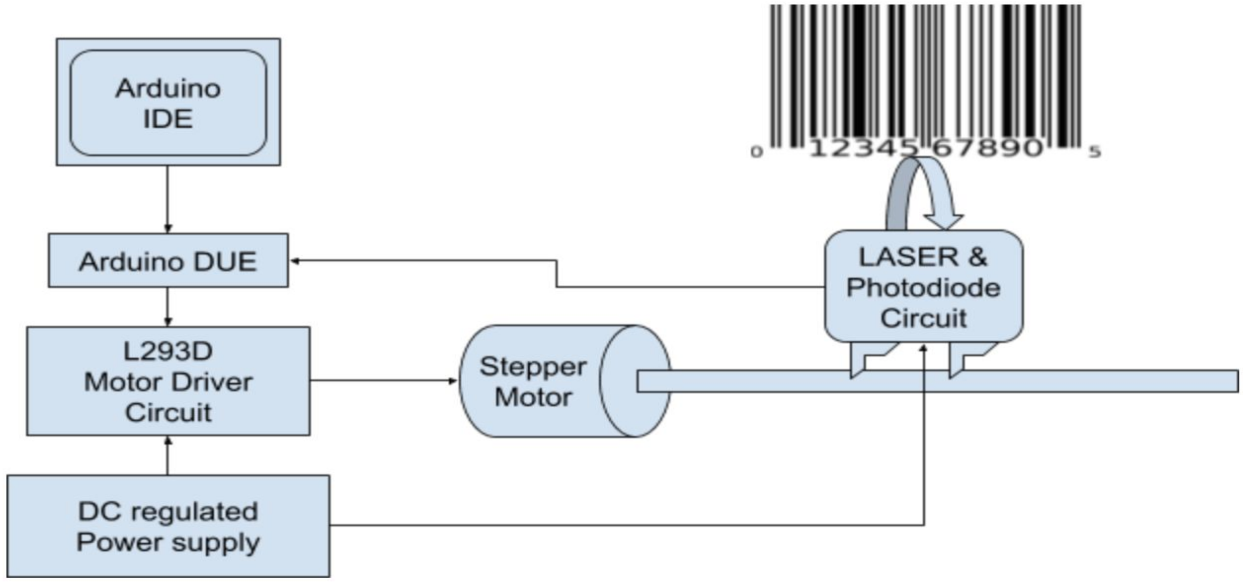
Table of Contents

Chapter 1: Introduction

Objective

1. To design and develop a Barcode Scanning Engine.
2. Decode a bit stream generated by the engine to get the actual decoded data.
3. Sending the data to a PC using serial communication via USB.

Block Diagram


Arduino IDE
0 12345 67890 5
Arduino DUE
LASER & Photodiode Circuit
L293D Motor Driver Circuit
Stepper Motor
DC regulated Power supply

Motivation for the Project

Currently, all barcode scanners that are commercially available in India are either imported from countries like China, making them unaffordable to the average shopkeeper/businessman, or are of low quality. We aim to make a “Made-In-India” barcode scanner which can be made available to the general public.

Chapter 2: Project Design

Barcode scanners are ubiquitous today in many commercial settings such as retail, pharmaceuticals, shipping etc. Their widespread use can be attributed to their simplicity compared to images as well as reliability and robustness to noise.

While methods based on cameras exist, and smartphones today also come with dedicated applications to do the same, cameras are much more expensive than simple laser and photodiodes. They also require dedicated focussing and good lighting conditions. Image-based decoding of barcodes is particularly susceptible to changes in lighting conditions and grainy noise in the acquired image. Similar dependence on ambient lighting plagues the use of CCD scanners. For all these reasons, laser based barcode scanners are still the preferred choice for many settings.

We have also hence decided to build a laser-photodiode based barcode scanner. We now proceed to detail our choices for the various subsystems in our setup.

1. Optical Subsystem

We chose to use a red laser due to their easy availability and as they are typically cheaper than say, a green laser. Furthermore, there are easily available photodiodes with a peak response close to red.

We procured the strongest red laser that we could, keeping cost constraints in mind. The spot size of the laser beam should be as small as possible, and should ideally be smaller than the width of the barcode bars (0.33mm). However, we were unable to procure such a laser, and instead accounted for this fact with more robust signal decoding.

We also tried acquiring a high quality lens or mirror to build a focussing system, but we could not get such lenses and mirrors from the various shops, labs and Professors that we asked. Objective lenses available online were also prohibitively expensive, with costs upwards of Rs.4000 (which is more than four times the cost of our entire current setup).

Another strategy that we explored was to try and reuse the optical subsystem in an optical CD drive (which as it is we were breaking down to harvest the linear actuation setup from). We found that the partially reflecting mirrors used in this setup were reflecting only for a small range of frequencies and hence we could get it to work with our laser. We also found it difficult to position our laser and photodiode to efficiently collect reflected light.

This unavailability of the right lenses and mirrors strongly influenced our design choices pertaining to the optical setup, and the system as a whole. It was one of the reasons for not attempting to replicate the system in commercial laser barcode scanners, that have very high quality, specifically engineered mirrors (the mirrors have a small wedge on a highly polished curved surface, something that would be very hard to build given the current resource constraints). Furthermore, while commercial barcode scanners use a mirror to concentrate reflected light from the barcode, without access to such high quality mirrors, we decided to come up with a system that collects as much reflected light as possible without the use of lenses or mirrors (this is also because we are using an off the shelf photodiode, with a considerably large dark current).

This is what inspired us to build a system in which the photodetector is always directly above the barcode. It should also be noted that in our setup, the photodiode is always directly in the path of the reflected light. This ensures that the maximum amount of reflected light is collected (assuming reasonable smoothness of the reflecting surface). Another very important advantage of our setup is that the amplitude of peaks and troughs of the recovered signal do not depend on the position along the barcode, as the angle of incidence and reflection are always fixed unlike in the case of the commercial barcode scanner (in which case the peaks towards the ends of the barcodes would be smaller, and the variation in general would be nonlinear). This makes decoding much more reliable and simpler to implement.

2. Mechanical Subsystem

We have mounted our laser and photodiode setup on a platform that is moved across the barcode by a stepper motor, using a linear-screw actuator. A smooth screw with precisely spaced grooves (ie the pitch is constant along the length of the screw) has been mounted on a stepper motor and the platform is coupled to the screw via a spring-loaded clamp.

This setup is radically different from that found in commercial scanners, that have a mirror attached to a small magnet and mounted on a pivot which is made to oscillate by an inductor. This setup is much more compact than our setup, allows for faster scanning and will also consume less power. However the inductor, small magnet and pivot are all very delicate and hard to build due to the precise nature of the setup as well as the materials required. These components also need to positioned very precisely. Furthermore, this setup leads to a lot of complications in decoding: The interaction between the inductor’s magnetic field and the mounted magnet is non-linear. The laser being projected on the barcode varies as $\cos(\theta)$, where θ is the angle made by the moving mirror (with the mean position), another non-linearity, leading to distortion in the widths of the bars. Also as mentioned in the previous section, the angle of incidence of the laser varies with time and hence so does the amplitude of the observed peaks. We decided not to attempt building such a setup because it would be very hard to complete within the span of this course.

We do away with all of these non-linearities in our setup by using a reliable stepper motor along with a precise linear actuation setup harvested from an optical CD drive. It is easy to ensure a uniform scan rate, with no position-dependent nonlinearities.

In the course of coming up with this setup, we had also considered other methods of linear actuation. A rack and pinion setup was first considered. We were unable to mount the gear onto the shaft of the motor without introducing an offset with respect to the axis. This is highly undesirable as this would lead to non-uniform sampling. Further, the size of the setup and the weight of the motor used was also a concern. A belt and gear setup was also considered. This idea was rejected due the challenge of ensuring proper tension in the belt, excessive friction, as well the unnecessarily bulky setup. It is also to be noted that the final setup that we chose to go with allows for much more precise control using a smaller motor and more compact setup than the other two methods mentioned above.

3. Signal Decoding

[4] provides a good theoretical background on how the laser spot width affects decodability. They proposed a mid-point estimation based method that uses an approximation of the inverse filter for deblurring. This method requires an empirical estimation of the laser spot width. In [3] they perform minimise least square error via gradient descent. In [1] they minimise least square error using a matrix-computation

based approach, using some additional statistical assumptions and use the structure of the UPC A convention to decode more accurately.

Finally in [5] the authors propose an effective method based on the zero crossings of the second derivative. Our decoding algorithm is based on [5], with some additional code to cover corner cases. We have not implemented the edge enhancement filter which is an approximate inverse filter proposed in [5], which may give further improvements.
This method was chosen due to simplicity of implementation and due to its theoretical appeal. The computational complexity of this approach is also much lower as it only uses the points of extrema in computing bar widths. It also does not require any empirical estimates of SNR or laser spot width.

References:

1. Dariusz Madej, Reversing Convolution Distortion in a Laser Bar Code Scanner, Automatic Identification Advanced Technologies, 2007 IEEE Workshop
2. Mark Iwen, Fadil Santosa, Rachel Ward, A symbol-based algorithm for decoding bar codes, arXiv:1210.7009
3. Todd Wittman, Deblurring and Restoration in Barcode Signal Processing, Siam Conference on Image Science 2004
4. Eugene Joseph and Theo Pavlidis, Bar Code Waveform Recognition Using Peak Locations, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 16, NO. 6, JUNE 1994
5. Stephen J. Shellhammer, David P. Goren, and Theo Pavlidis, Novel Signal-Processing Techniques in Barcode Scanning, IEEE Robotics & Automation Magazine, March 1999

4. Barcode Convention

Barcodes were devised as a graphical machine readable representation of the **UPC,** which stands for the Universal Product Code.The original UPC code is also known as UPC-A or UCC-12, the latter to indicate its 12-digit size. (UCC is an abbreviation for the Uniform Code Council). Since every digit in the barcode is encoded with 7 bars/bits, the total number of bars required for 12 digits is very large, which makes the total width of the barcode considerably large for its standard size.
There is another encoding which is essentially a variation of UPC-A which allows for a more compact barcode. UPC-E barcode resulting after conversion from a UPC-A barcode is about half the length as the UPC-A barcode.

In choosing a barcode convention the criterion that we chose were:

1. The convention should be universal and widely used.
2. The density of digits encoded per bar should be high.
3. The total number of bars used must not be too large.

Criterion 1 is to allow our system to be used widely while 2 and 3 are a consequence of finite length of the screw in our linear actuation system. Our platform can move up to 4 cm and this is then the maximum size of a barcode that we can scan. As the number of bars in a barcode convention increase, this will mean that if this barcode is packed within 4 cm, the width of each bar will be small, leading to difficulties in decoding due to large spot width of the laser.
Hence we chose the UPC E convention over UPC A, ITF 14, EAN 8 etc.

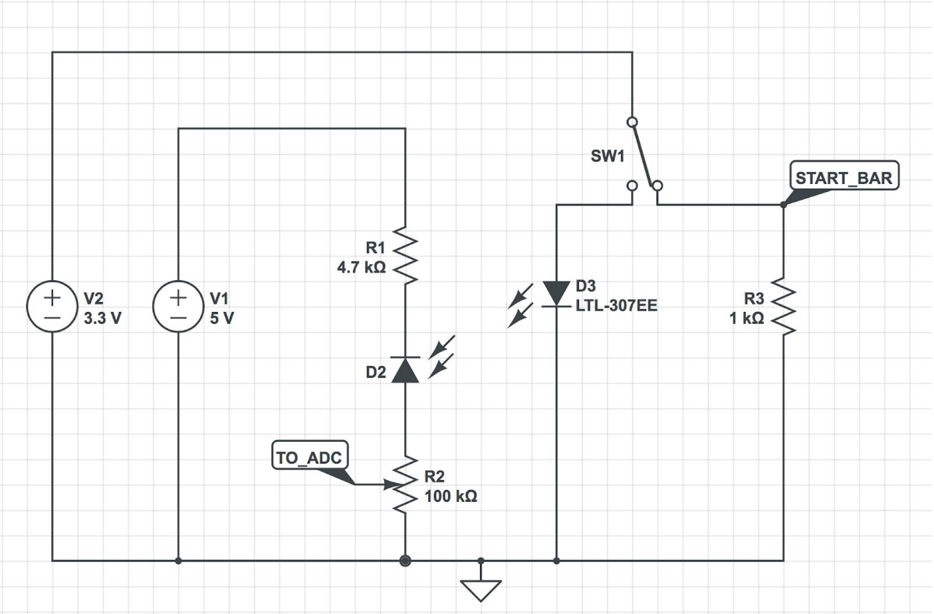
Chapter 3: Project Implementation

We have implemented a linear-scanning based laser barcode scanning engine that uses a laser-photodiode pair, with a stepper motor driven linear actuator harvested from a optical CD-drive to move the sensor setup across the barcode.
We now proceed to describe the subsystems of our setup.

1. Optical-Electrical Sub-System

Our Optical-Electrical sub-system consists of a microcontroller (Arduino DUE), a LASER and a photodiode. Additionally, we also have a circuit to control the linear actuator.
The Laser is powered by our Arduino via its 3.3V supply pin. It is pointed at the barcode and a photodiode has been placed in the path of the reflected light. When the laser is pointed at a ’white’ bar, the light is strongly reflected, resulting in a higher current passing through the receiving photodiode circuit. Similarly, when the laser is pointed at a ’black’ bar, the light is not reflected or weakly reflected, resulting in very little current passing through the receiving photodiode circuit.
The photodiode is reverse biased and acts as an optical sensor, and the current through it is passed via a resistor to generate a voltage. This voltage is then periodically sampled via our microcontroller ADC, generating a time-evolving signal of the reflected

light as the platform is moved across the signal. This signal is then processed to decode the final output.

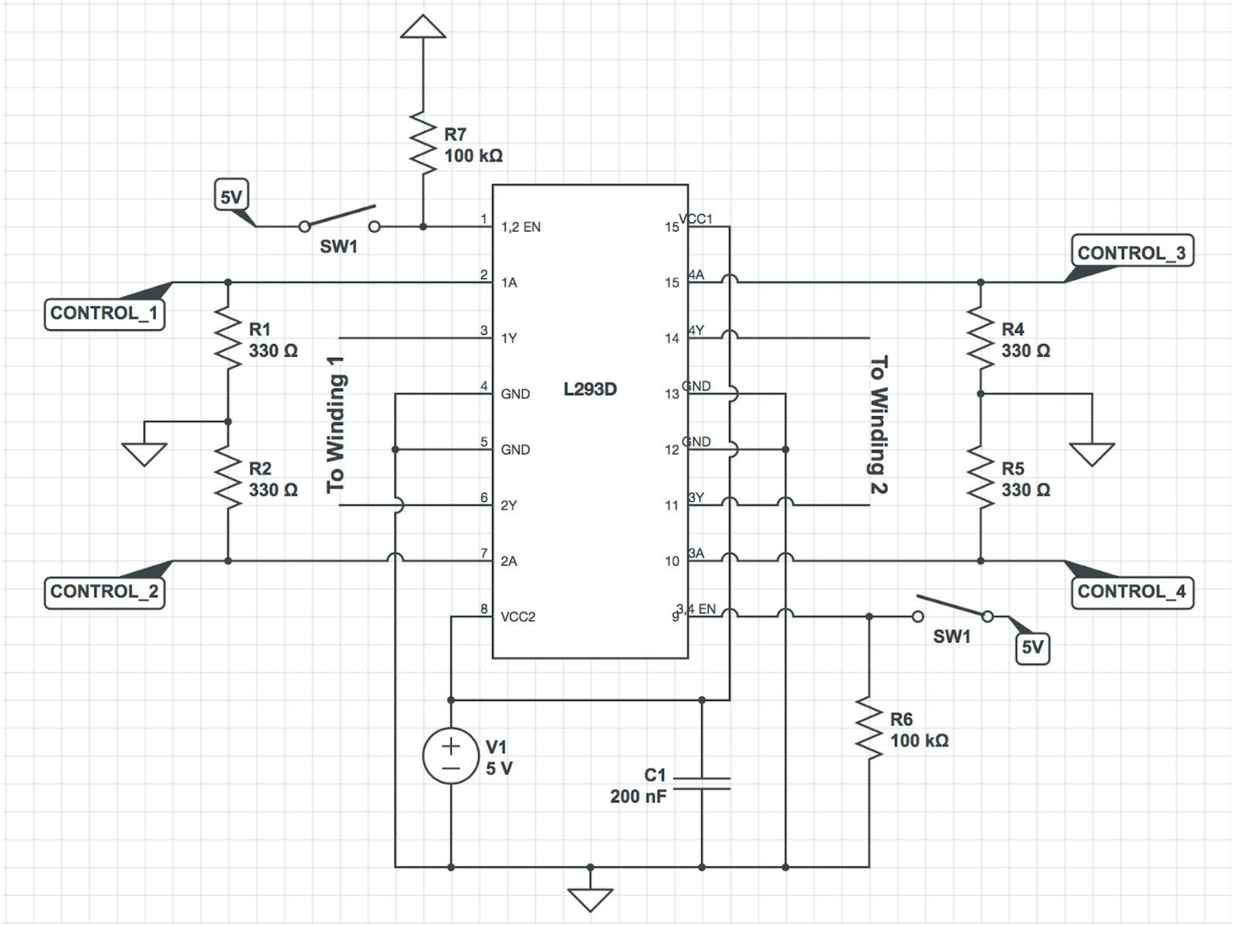


In the circuit shown above, the reverse-biased photodiode (D2) generates a current which is converted to a voltage by R2. We have chosen a variable resistor to account for different lighting conditions and to limit the peak voltage that is sampled by the ADC of our Arduino DUE. This is because saturation is highly undesirable for our decoding algorithm. R1 has been placed as a method of verifying whether the photodiode is placed with the correct polarity or not. If the photodiode has been placed correctly, the potential at the n-terminal of the photodiode will reduce when light falls on it (due to the voltage drop across R1). If, however, it's been incorrectly oriented, the potential will be constant (approx 5V due to R1/R2 ratio) irrespective of the light falling on it.
A switch has been placed to start the system up by turning on the laser. The node 'START_BAR' then switches from 3.3V to 0V. This is detected by the microcontroller which then starts sampling the data.

2. Mechanical Sub-System

Our Mechanical sub-system consists of a Linear Actuator, and a platform with the sensor circuit mounted on it, that is to be moved across the barcode.
The actuation is done by a stepper motor, with a shaft connected to its rotor. The shaft has wide and smooth threading like grooves. The platform is restricted to move along only one horizontal direction as it is mounted on a a pair of parallel smooth metallic rails.

To couple the platform to the rotating shaft there is a spring-loaded clamp protruding from the moving platform with teeth that fit into the grooves of the shaft. The platform is thus pushed along when the shaft rotates. The direction of rotation of the shaft determines the directions of linear motion of the platform.



3. Signal Decoding

Our decoding method is based on the paper Novel Signal-Processing Techniques in Barcode Scanning by Stephen J. Shellhammer, David P. Goren, and Theo Pavlidis (IEEE Robotics & Automation Magazine, March 1999). We have further implemented some checks to ensure correct decoding.

A barcode can be thought of as being a collection of spaced rectangles, and hence as a sum of shifted step functions. Hence the recovered signal is also expected to have sharp transitions. Furthermore, its derivative is ideally a set of deltas. We can detect edges by simply observing that they are the points of maxima and minima of the derivative of the signal.

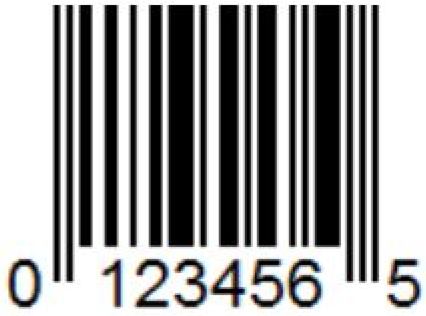
However, the laser has finite spot width which is often comparable to, if not greater than (as is true in our case) the width of the smallest bar. This means that the signal actually retrieved is the convolution of the original barcode with the spatial response of the laser. The most widely used assumption, that has also been used in the above paper is that the spot of the laser has a Gaussian profile. From this we can conclude that the derivative is a sum of shifted Gaussians.

As long as the Gaussians are spaced far enough from each other, the maxima and minima are preserved. But if they are too close, then the detected peaks may not always correspond to the peaks of the Gaussians. Hence, the spot width of the laser fundamentally limits how thin the bars of a decodable barcode can be.

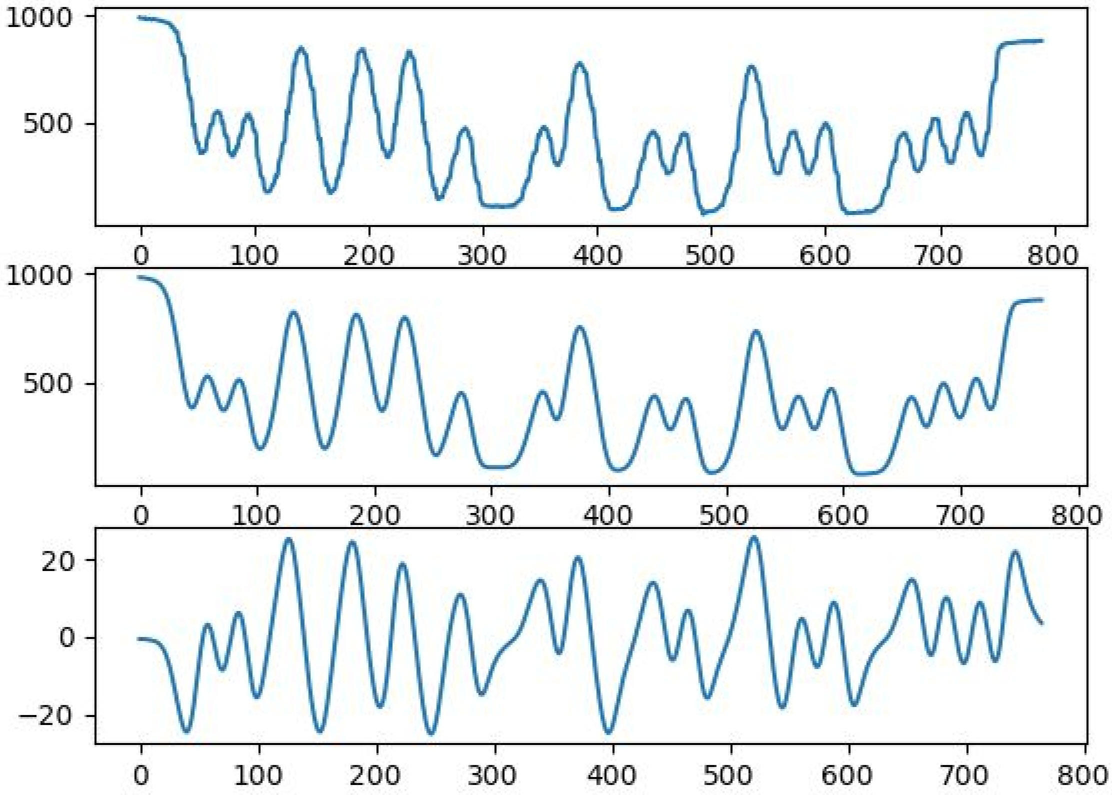
The pseudocode for the algorithm implemented is:

1. Smoothen signal acquired from photodiode using a moving window average (window length of 5 was chosen and the data was smoothened 5 times using this window).
2. Compute a new signal which is the numerical derivative of the above signal.
3. Smoothen this new signal if necessary.
4. Compute the maxima and minima of this derivative signal.
5. Discard extrema whose magnitude is small (noisy peaks).
6. Discard saddle points that have been misclassified as extrema.
7. Discard all but one extremum when there are many extrema very close to one another.
8. Compute the widths of black and white bars as the distance between these alternating maxima and minima.
9. Group the 24 relevant widths into 6 groups of 4 each (which corresponds to the 6 encoded digits), normalise and round off to compute the estimates for the barcode widths. Note that this method of locally grouping and then normalising helps improve tolerance to non-uniformity in sampling if any.
10. Correct errors using parity checks.
11. Use these bar widths to decode the signal using the chosen barcode convention, in this case UPC E.

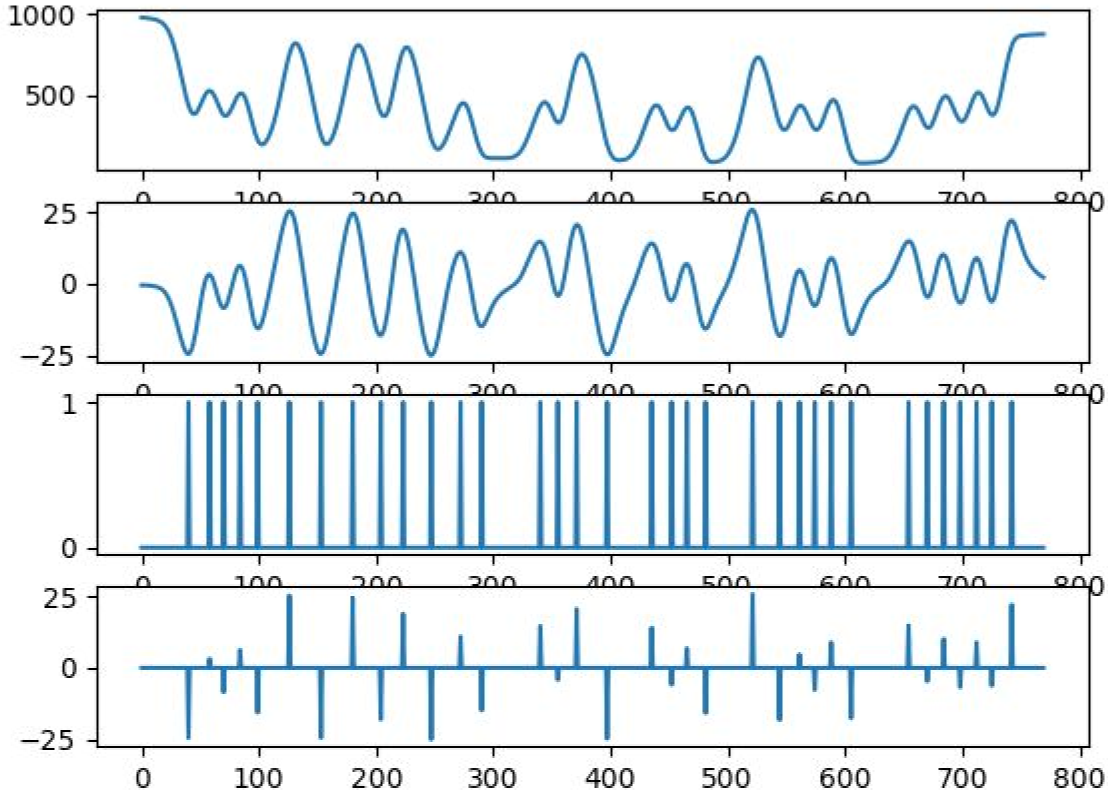
Our decoding steps are illustrated through a sample run on a randomly chosen barcode that follows the UPC E convention. The chosen barcode is:



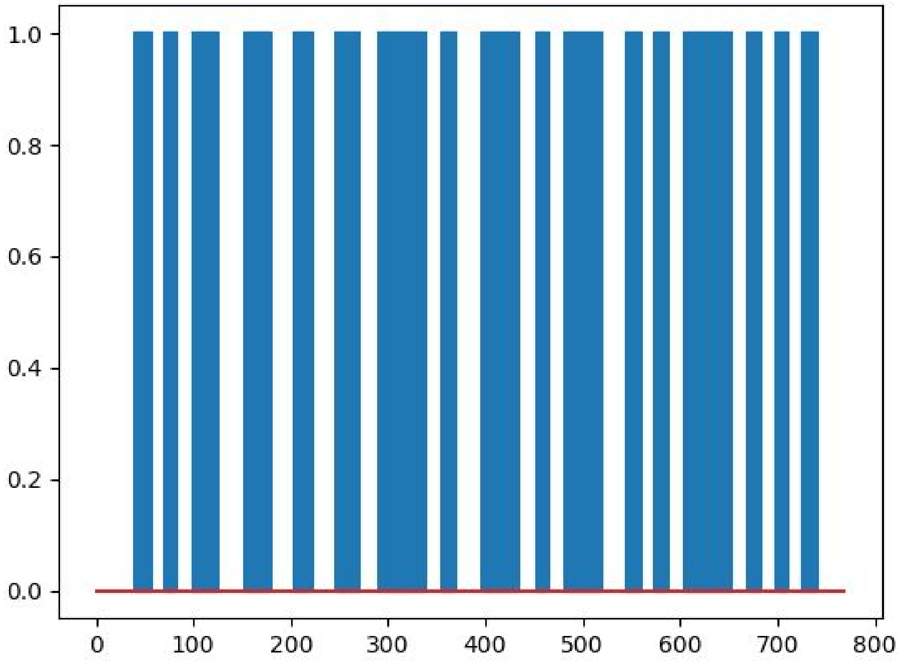
In the following plots, the first plot (from the top) is the signal received from the photodiode. The second plot is the smoothened version of that signal. The third plot is the smoothened derivative of the smoothened signal.



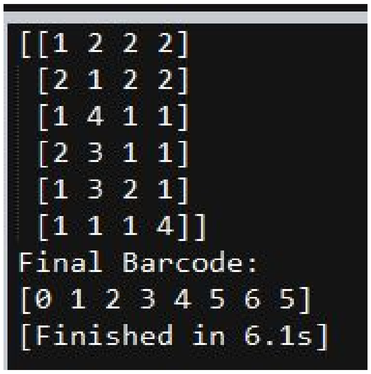
In the plots below, the first plot is the smoothened signal and the second plot is its smoothened derivative. The third plot is the locations of the detected extrema. The fourth plot depicts the magnitude of the derivative at these extrema.



The following figure depicts the reconstructed barcode:



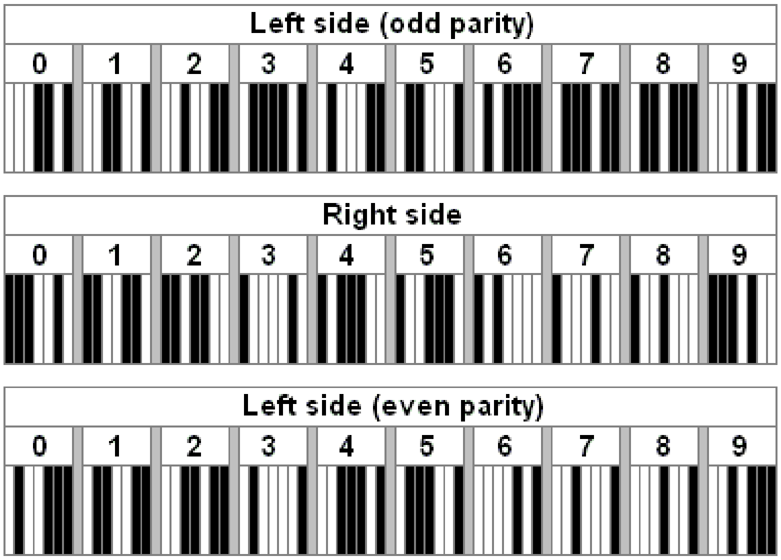
And the final, correctly decoded barcode is:



```
[[1 2 2 2]
 [2 1 2 2]
 [1 4 1 1]
 [2 3 1 1]
 [1 3 2 1]
 [1 1 1 4]]
Final Barcode:
[0 1 2 3 4 5 6 5]
[Finished in 6.1s]
```

4. Barcode Decoding

In a barcode, each digit of an encoding is represented by seven black or white bars of equal size. Visually, two or more adjacent black bars appear as a single wide bar; white bars appear as space separators (also of varying size) between the black bars. If we call such wide (black or white) bars *generalized bars*, then, as a rule, a digit's representation always consists of four such generalized bars, which by necessity, interlace: either white-black-white-black or black-white-black-white.



An UPC-A code consists of twelve digits that are split into six left and six right digits. The digits in the two groups are barcoded differently. To barcode the first six (left) digits we use the "Left side (odd parity)" representation. For the next six (right) digits, use the "Right side" representation. One can easily observe that the difference between the two is in exchange of colors: white bars in one are black in the other, and vice versa. All twelve digits are represented by seven bars each.

Besides the aforementioned fact that every digit's barcode consists of four bars, we may make additional observations:

1. First of all, left side barcodes of either parity are of the form white-black-white-black and thus end in a black bar. The right side barcodes are of the form black-white-black-white and end in a space.
2. No (generalized) bar nor space is wider than four (regular) bars.

Any barcode starts and ends with three "guard bars": black-white-black. Barcodes corresponding to "two-halves" encodings, like the ***UPC-A*** and ***EAN-13***, also feature a middle guard in the form white-black-white-black-white.

UPC-E is a compact variation of UPC-A which removes all extra digits, and reduces actual information encoded in 10 characters into 6 characters. UPC-E has 6 digits encoded into the bars. Additionally there is a digit at the starting which tells the number system used for encoding the 6 digits, and there is a check digit at the end which is encoded in the parity of the six characters.

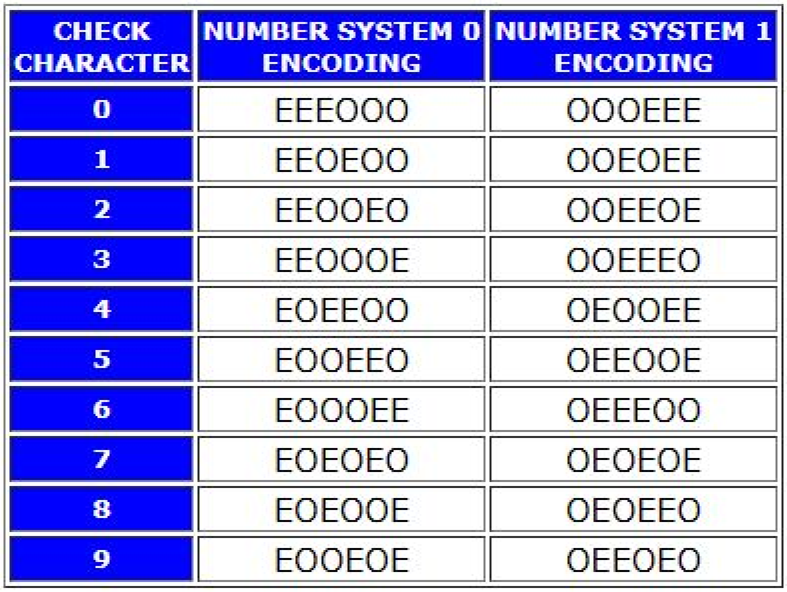
Structure

An UPC-E barcode has the following physical structure:

- Left-hand guard bars, or start sentinel, encoded as **101**.
- Six data characters, encoded from the parity table above.
- Right-hand guard bars, encoded as **010101** (a center-guard bar pattern with a trailing bar).

Encoding of characters

The characters are encoded with odd and even parity left handed encoding of the EAN-13 character format. The encoding (even OR odd) of each digit is decided by jointly by the number system and the check digit. Thus these 6 digits encode the entire string of length 8.



CHECK CHARACTER	NUMBER SYSTEM 0 ENCODING	NUMBER SYSTEM 1 ENCODING
0	EEEOOO	OOOEEE
1	EEOEOO	OOEOEE
2	EEOOEO	OOEEOE
3	EEOOOE	OOEEEO
4	EOEEOO	OEOOEE
5	EOOEEO	OEEOOE
6	EOOOEE	OEEEOO
7	EOEOEO	OEOEOE
8	EOEOOE	OEOEEO
9	EOOEOE	OEEOEO

Decoding

The actual process of decoding is the reverse of encoding. First the 6 digits are decoding from their 7 bar representation. Then parity is found for every digit from the number of black bars out of the 7 bars, as the Odd parity encoding has odd number of Black bars while even parity encoding has even number of Black bars. Finally, from the parity of each digit the number system and the check digit can be extracted as can be observed from the table shown above.

References:

1. https://www.cut-the-knot.org/do_you_know/BarcodeEncoding.shtml
2. https://www.cut-the-knot.org/Curriculum/Arithmetic/UPC.shtml
3. http://www.barcodeisland.com/upce.phtml

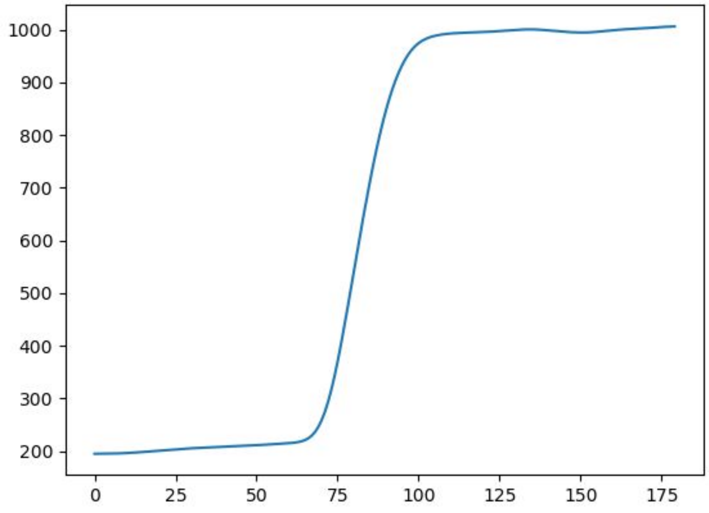
Chapter 4: Performance Evaluation

System Characterisation

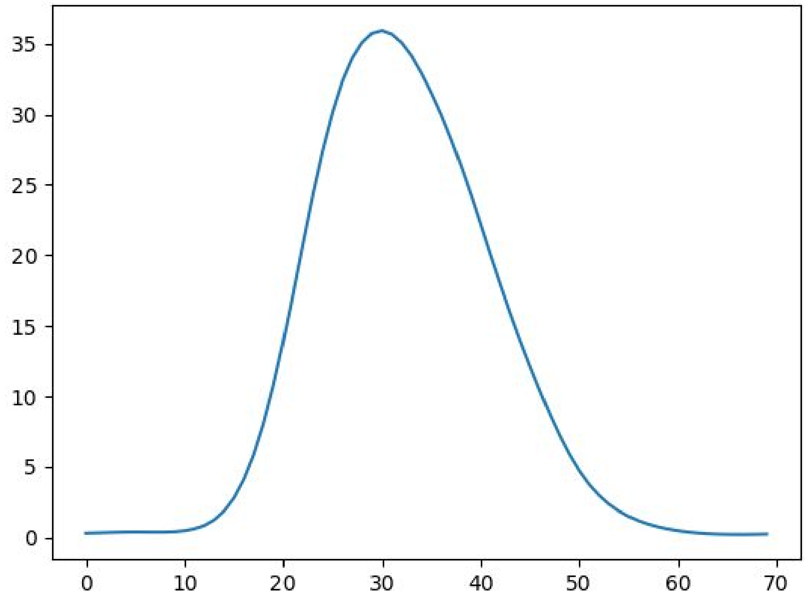
We characterised the various components used in our setup:

1. Photodiode dark current: **110 nA**
2. Laser current : **18mA** (drawn at 3.3V, with an unknown internal series resistance)
3. Current drawn by DC motor :
   a. Not operating: **26 mA**
   b. During operation
      i. Average : **144 mA**
      ii. Max : **175 mA**
4. Tilt of laser from vertical axis : **18.8 degrees**
5. Pitch of screw in linear actuator : **0.302 cm** (13.25 revolutions for 4 cm moved)
6. Total scan and decoding time: **5s**

We also conducted a simple experiment to determine the spatial response (spot shape) of the laser. We simply recorded the values from the laser being moved from a black to a white bar (with no other bars on either side). This is equivalent to convolving the laser's response with a step function. The recorded signal looks like:



Hence by differentiating this curve, we can retrieve the spatial response of the laser. The retrieved response looks like:



It can be seen that the response does indeed look like a Gaussian, as is assumed in most of the literature. Having estimated this response, we can fit a Gaussian to it and estimate the variance. This can be used to perform approximate inverse filtering, as described in the literature (refer Section 3 of Chapter 2). Note that naive inverse filtering using the reciprocal of the FFT does not work, as we verified in code. This is because the DFT of a Gaussian is also Gaussian and hence the inverse filter in the frequency domain is of the form e^{x^2} (with x scaled appropriately) which is a high pass filter and hence amplifies high frequency noise as well, leading to a highly corrupted recovered barcode.

Problems faced

- **Stringent LASER requirements:** Based on the size of a standard barcode and the number of bars per barcode, we needed a laser of a spot size less than 0.33mm. Also, the LASER needed to be powerful enough to reflect off a barcode and still have enough energy to cause a swing at the receiver. We spoke to manufacturers, shopkeepers and Professors to try and find such a laser, but we couldn’t procure one. We then had to do with a LASER of lower quality with a spot size larger than the thinnest bar, which resulted in interference due to neighbouring bars as well as lower swings. By implementing a more robust decoding algorithm, we were able to overcome this disadvantage.

- **Optical components:** As mentioned in Chapter 2, we were unable to acquire good quality lenses and mirrors to focus the laser beam and to collect the reflected light. We overcame this limitation by designing a setup that collected as much of the reflected light as possible.
- **Mechanical Actuation:** To implement our barcode scanner to get hundreds of samples across a distance of 3-4 cm, we required an accurate motor with good precision. We considered servo and stepper motors. We also needed a linear actuator that could be driven by this motor. We decided to open up a DVD Drive as a way to implement our optical system during which we came across a small 4 pole DC motor with a a fitted linear actuator fitted right inside! The platform attached had a linear movement of 4 cm which was perfect for our requirement

Lessons Learnt

LASER Operation: During the course of this project, we have either completely or partially burnt 5-6 lasers. We now give 2 main lessons that were learnt from this:

1. Power Supply: The voltage fed to the LASER must be as constant as possible, and must not exceed 5V. The power supply that we were using initially had a large transient (close to 30V) when it was switched on. This transient led to the laser being damaged. We concluded that this was indeed the problem by building a voltage regulator circuit and observing that the laser did not get damaged in this scenario. Having determined this, we then powered the laser from the 3.3V port on the Arduino (after having ensured that the Arduino can provide the required current to the laser).
2. Heat Effects: Heating the terminals of the LASER by either soldering wires onto the terminals, or by the application of hot glue from a glue gun on the outer surface of the laser (to attach the laser to our moving platform) can also damage the laser.

Signal Processing: Our decoding algorithm requires us to differentiate the signal obtained from the photodiode. This signal is noisy, and hence the derivative will be even noisier (differentiation amplifies high frequency components, and hence high frequency noise). The signal as well as its derivative can be smoothened to overcome this problem. However, the more aggressive the smoothening, the more the signal also gets distorted. This is particularly troublesome because this means that the detected bar widths will also change, and this can lead to incorrect decoding.

We found that smoothening the signal more aggressively instead of its derivative works better, and preserves the bar widths more closely.

Furthermore, we found that using a window of fixed length and equal weights within the window, when used to smoothen the data with multiple passes works better than exponential smoothing (but is more computationally expensive).

Chapter 5: Conclusions and Future Work

In this project, we have built a linear-scanning based laser barcode scanning engine that uses a laser-photodiode pair, with a stepper motor driven linear actuator harvested from a optical CD-drive to move the sensor setup across the barcode. The system is able to successfully decode UPC E barcodes.
The total cost of the project was under Rs 1000, which is more than three times less than that of most commercial barcode scanners. The primary limitation of our setup is slow speed of decoding and relatively larger setup. But we believe that the reliable decoding offered at the much cheaper price makes this a competitive option.

In the future, one possible improvement to the decoding algorithm is to use an edge enhancement filter, which is essentially an approximate inverse filter, estimated using the variance of the Gaussian spot shape of the laser (which we have measured empirically). Currently, our setup has wires going from the laser to the Arduino as well as from the photodiode to the Arduino. These wires can be replaced by conducting rails, making for more compact implementation. These wires can also be a source of unwanted tension and hence can lead to non-uniform velocity if the wires are compressed too much, We can also try to power the stepper motor from the Arduino itself (something we did not risk for fear of damaging the Arduino). We could also make on integrated board to contain all the electrical components used in our system.