MTG Analyser

Sebastian Pech

13 March 2013

Abstract

This document describes the process of developing a automated card collection analysing system.

1 Concept

Over the years the author collected cards of the famous trading card game **Magic: The Gathering**.

> Magic: The Gathering (MTG; also known as Magic) is the first trading card game, created by Richard Garfield and introduced in 1993 by Wizards of the Coast. [...] Magic can be played by two or more players each using a deck of printed cards[...].
>
> An organized tournament system and a community of professional Magic players has developed, as has a secondary market for Magic cards. Magic cards can be valuable due to their rarity and utility in gameplay.[1]

At the end of 2012 more than 20.000 different cards exist.[2]

With increasing size of the authors collection it became difficult to keep an overview of all cards. Older editions did not include colored symbols for the rarity therefore many knowledge of the different cards is necessary. Lastly the collection is spread across multiple boxes, folders and stacks. For this reasons the author wanted to create a system to automatically archive all cards.

In the first chapter the **design** of this system will be described. The whole system covers the analyzing software, hardware to sort the cards and a frame for the hardware. The frame must be easy to build preferable with material that already exists in the authors household. Therefore **Lego** parts are used for this purpose as described in the design chapter. The software and hardware is specified in the following chapters.

After designing the system the **building** and assembly process is documented. The Software chapter contains the programming the **microcontroller** and the analyzing software. Lastly the conclusion of the project is attached.

[1] `http://en.wikipedia.org/wiki/Magic:_The_Gathering`

[2] `http://gatherer.wizards.com/Pages/Default.aspx`

2 Design

The basic idea of the system contains several areas as seen in Figure 1. First a container for the card pile is required. From this container the cards have to be moved on a conveyer belt. Throwing the cards on the conveyor belt is done with a motor connected to a wheel. During the test phase special attention must be put on the motor control to avoid throwing multiple cards at once.

The conveyor belt moves the card from the drop-off location to a bin. During the movement a Webcam keeps track of the card. The Webcam takes a picture of the card and tries to match it with an reference image. If the card was found the name of the card is saved otherwise the image itself is saved for manual post-processing.

To optimize the image tracking process serveral LEDs light up the card. For keeping track of the cards position a photoconductive cell might be used. This could also be improved by adding an IR-LED and an IR phototransistor to create a IR barrier.

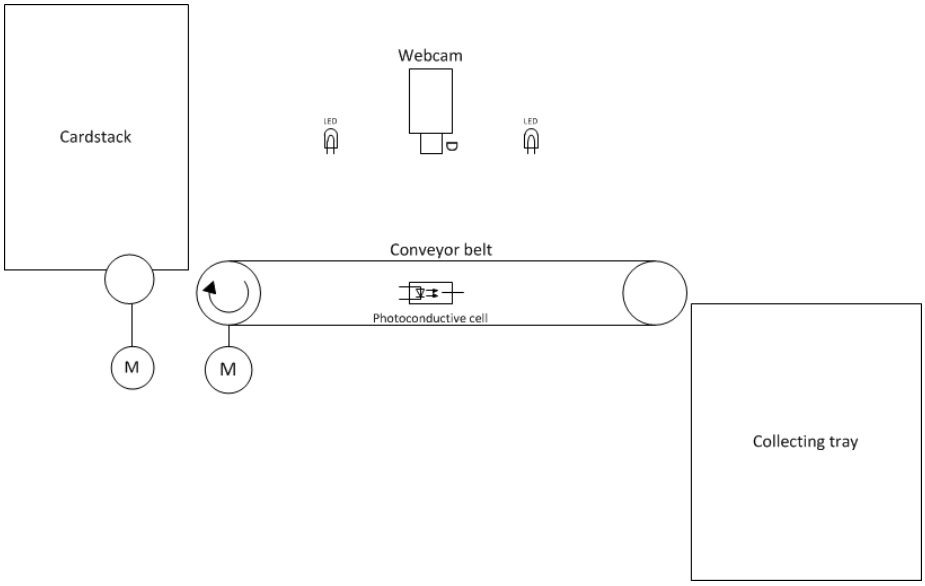

Figure 1: Plan

3 Software design

The software for this project is called MTGAnalyser and its main purpose is taking images from a webcam and try to find the cards seen in the image. However, for further convenience the analysis of photos should be possible to allow usage of this software for users which do not own webcams or want to

take pictures with an DSLR or mobile phone.

3.1 Structure

MTGAnalyser contains a main window which allows access to several functions that are divided into three areas. Input, Processing and Output. The functions and areas are described below.

3.1.1 Input

Within the input area all functions for providing image data to the image processing part is encapsulated.

Image Clicking the image buttons allows the user to save/copy an existing image to the work folder and update the "'View Images"' dialog.

Webcam The Webcam window allows the user to select a video source and start the streaming. For analysing and displaying of the image the "'Check Images"' will be opened.

Robot The Robot dialog uses the same window as the webcam functionality. Only special options for controlling the hardware described in this project are shown. This function is not relevant for the common user of the software.

3.1.2 Processing

Processing is the main area of the software and divided in the area for viewing and processing images.

View images The view images windows displays a list of all copied and webcam generated images for selection in the "'Check Images"' dialog.

Check images This is the core of the application. This window displays the input image from the image viewer or the webcam and tries to find the cards. All cards are listed and the user can check the information. Furthermore all found cards are stored internal for saving them to a file. The analysing process is described in the chapter Functions.

3.1.3 Output

The found images can be saved in different output formats. The formats are Comma-separated values (usually with .csv extension), the deck format (.dec) which is used around the Magic scene quite often and the library format of the authors software **Archivist 2**.[3]

[3] `http://www.spech.de/node/45`

CSV A comma-separated values (CSV) file stores tabular data (numbers and text) in plain-text form. Plain text means that the file is a sequence of characters, with no data that has to be interpreted instead, as binary numbers. A CSV file consists of any number of records, separated by line breaks of some kind; each record consists of fields, separated by some other character or string, most commonly a literal comma or tab. Usually, all records have an identical sequence of fields.[4]

CSV files make it easy to work with them. Many programming languages contain libraries to read the format and almost all spreadsheets programs support the format.

The software will output the data in the format:
Amount;Name;Edition;Rarity;(Carriage return, linefeed)

Deck The Deck format is a human readable and writable format which is supported in most Magic software. The basic format contains only the amount and the name of a card per line. These values are separated by a blank. Comments can be inserted by using slashes and cards in the sideboard are prefixed with "'SB: "'. For this software the basic format is sufficient.

The software will output the data in the format:
Amount Name(Carriage return, linefeed)

Archivist 2 Library Archivist 2 is a program created to search card, manage the library and create decks. Statistic tools support the creation of the decks.

Archivists library format is very simple but very powerful. Every card is stored with the multiversid and the amount separated by a semicolon. The id is equal to the Gatherer[5] Website created by the distributor of Magic cards. With the id all information of a card can be read (Name, text, edition, rarity, flavor text, orcle text, ...).

The software will output the data in the format:
Id;Amount(Carriage return, linefeed)

3.2 Functions

Analysing the image to find a matching card is separated into multiple subprocesses. First the source image is searched for the card or cards regions. Every region will be extracted and hashed. After that the hash is compared to the hashes of every existing card reference image. If the difference between the hashes is small enough the card is found otherwise the source image is saved for manual editing.

[4] `http://en.wikipedia.org/wiki/Comma-separated_values`
[5] `http://gatherer.wizards.com/`

3.2.1 Find card Images

Finding the cards within the image is a job that perfectly fits the AForge.NET Framework.[6]

> AForge.NET is an open source C# framework designed for developers and researchers in the fields of Computer Vision and Artificial Intelligence - image processing, neural networks, genetic algorithms, fuzzy logic, machine learning, robotics, etc.

For using the shape checker class[7] the image has to be prepared. First the image will be converted with filter collection.

1. Graysacle with BT709.[8]
2. Sobale edge detector.The filter searches for objects' edges by applying Sobel operator. Each pixel of the result image is calculated as approximated absolute gradient magnitude for corresponding pixel of the source image[9]
3. Threshold binarization. The filter does image binarization using specified threshold value. All pixels with intensities equal or higher than threshold value are converted to white pixels. All other pixels with intensities below threshold value are converted to black pixels.[10]

After the filtering the resulting image is searched for rectangles that might be cards.

3.2.2 Generate and compare hashes

For every existing card a image hash must be build. Than for every found card in the source image a hash must be build in the same manner. The hash algorithm has to be fast and must handle modifications (color, stretching, rotation, ...) in order to be suited.[11] Then the distance of all hashes has to be build. If the difference between the hashes is small enough the card is found.

4 Hardware design

The Hardware design must cover three different functions. First the LEDs for lighting up the scene must be turned on and off, secondly the motors for moving

[6] http://www.aforgenet.com/framework/

[7] http://www.aforgenet.com/articles/shape_checker/

[8] http://www.aforgenet.com/framework/docs/html/f0c2a0fb-12a4-0ab4-81cd-095f482b1383.htm

[9] http://www.aforgenet.com/framework/docs/html/2c8218cc-921c-34d8-5c88-39c652488490.htm

[10] http://www.aforgenet.com/framework/docs/html/503a43b9-d98b-a19f-b74e-44767916ad65.htm

[11] http://www.hackerfactor.com/blog/index.php?/archives/432-Looks-Like-It.html

the conveyor belt and throwing the cards on the belt have to be controlled. Lastly but optional the card position can be tracked by a IR-barrier.

The basic circuit is shown in Figure 2. For the controlling of the LEDs the Arduino microcontroller is used.

> Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments.
>
> Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software running on a computer (e.g. Flash, Processing, MaxMSP).[12]

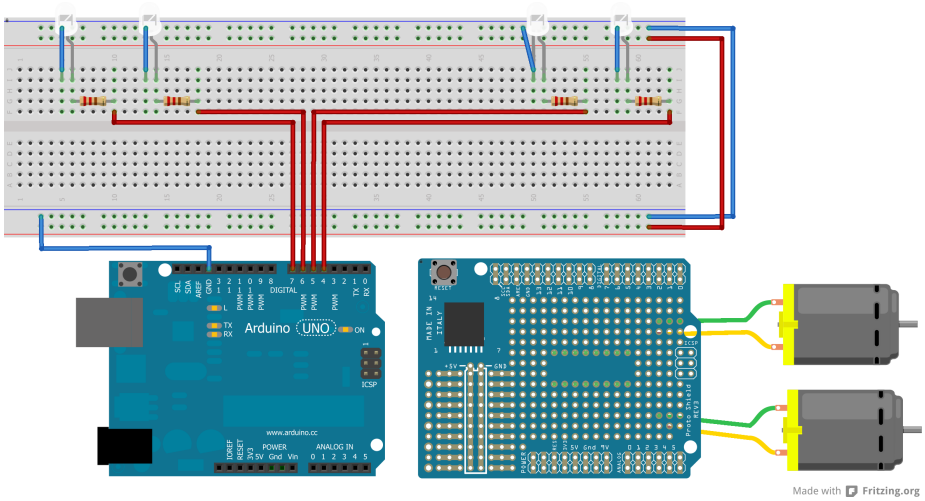

Figure 2: Circuit (Motor Shield as placeholder

For controlling the motors the Arduino is extended with the Arduino Motor Shield. Instead of building the Motor control with MOSFET H-Bridges[13] [14] this is an easier and faster to use approach.

> The Arduino Motor Shield is based on the L298, which is a dual full-bridge driver designed to drive inductive loads such as relays,

[12] http://arduino.cc/
[13] http://www.seas.upenn.edu/~ese206/labs/MOSFET/MOSFETHBridge.pdf
[14] http://garagelab.com/profiles/blogs/tutorial-l293d-h-bridge-dc-motor-controller-with-arduino

solenoids, DC and stepping motors. It lets you drive two DC motors with your Arduino board, controlling the speed and direction of each one independently. You can also measure the motor current absorption of each motor, among other features.[15]

5 Lego Design

Lego (trademarked in capitals as LEGO) is a popular line of construction toys manufactured by The Lego Group, a privately held company based in Billund, Denmark. The company's flagship product, Lego, consists of colorful interlocking plastic bricks and an accompanying array of gears, minifigures and various other parts. Lego bricks can be assembled and connected in many ways, to construct such objects as vehicles, buildings, and even working robots. Anything constructed can then be taken apart again, and the pieces used to make other objects.[16]

Due to the fact that Lego bricks can be assembled in many ways it works perfectly as the prototyping environment for the system. The whole construct will be separated in two independent areas which are connected in the end. The first part is the deck box with the ejector and the second one is the conveyor belt.

5.1 Deck box

The deck box must be big enough to hold a few hundred cards off the same size and enclose them as good as possible. For easy removal of the cards by hand a manual remove should be build in. The bottom of the Box must contain free space to attach the motor with the rubberized wheel. During the testing phase the right angle of the throw out hole must be found.

5.2 Conveyor belt

The conveyor belt is build as two long walls connected on the bottom with plates. At the beginning and the end wheels are attached to move the belt. The belt itself is build from long paper stripes connected with glue.

6 Software development

The MTG Analyser software was written using C# and the .NET Framework 4.0. For image processing the AForge.NET library was used as described in the design.

In the first version the software was created to process static images taken by a normal camera (Figure 3). Later the ability to track cards in real time

[15] `http://arduino.cc/en/Main/ArduinoMotorShieldR3`
[16] `http://en.wikipedia.org/wiki/Lego`

was added (Figure 4). The idea of processing the images is loosely based on the idea of the MagicVision project[17]. However the technique for comparing the images is different form their approach. Instead of using perceptual hashes and the pHash library image hashes are used.

> What is a perceptual hash? A perceptual hash is a fingerprint of a multimedia file derived from various features from its content. Unlike cryptographic hash functions which rely on the avalanche effect of small changes in input leading to drastic changes in the output, perceptual hashes are "close" to one another if the features are similar.[18]

Image hashing is based on the avaerag hash algorithm provided by Dr. Neal Krawetz [19]. Using the library ImageHash[20] with methos by David Oftedal of the University of Oslo, Norway[21].

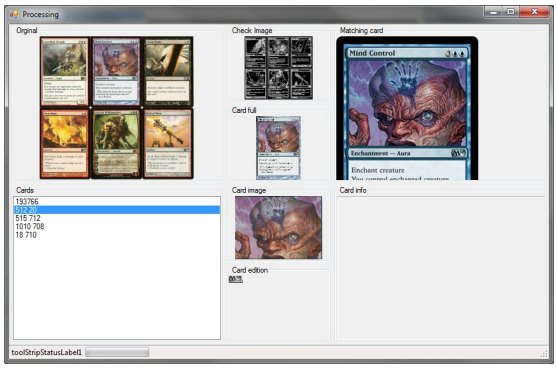

Figure 3: MTG Analyser - Image mode

For the analysing of the cards images using the Arduino the "'Roboter Mode"' was implemented. In this mode a small protocol was created to communicate with the Arduino using serial connections over USB. The Arduino software supplys drivers to connect to it using a serial device. As the .NET Framework contains classes for serial communication (System.IO.Ports) the implementation is very easy.

The serial connection protocol involves to request messages (PHASE1 and PHASE2) and two respones (P1OK and P2OK). The MTG Analyser sends "'PHASE1"' as soon as a new card can be processed. On receiving the message the Arduino throws a new card on the belt and moved it into the right sport. During Phase1 two LEDs are switched on. If the is supposed to be in place the

[17] `https://github.com/petesimard/Magic-Vision/blob/master/MagicVision/`
[18] http://www.phash.org/
[19] `http://www.hackerfactor.com/blog/index.php?/archives/432-Looks-Like-It.html`
[20] `https://github.com/jforshee/ImageHashing/blob/master/ImageHashing/ImageHashing.cs`
[21] `http://folk.uio.no/davidjo/computing.php`

Arduino replys with "'P1OK"'. The MTG Analyser takes the image using the webcam and processes it after that "'PHASE2"' is send to the Arduino. During Phase2 the card is moved from the conveyor belt to the collection tray. The "'P2OK"' response marks the end of the process.

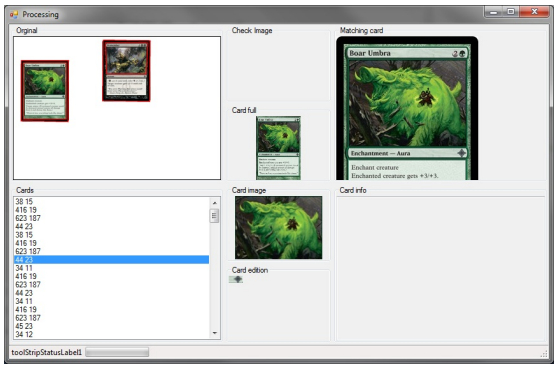

Figure 4: MTG Analyser - Webcam mode

7 Hardware

The hardware is assembled from the following parts.

- 1x Arduino Uno R3
- 1x Motor-Shield (Arduino)
- 2x Brushed DC Motor: 130-Size, 6V, 11.5kRPM, 800mA Stall
- 1x Breadboard 830 Pin
- 4x LED white 7150-18000mcd 20
- 1x Jumper Wires M/M 100mm
- 1x Jumper Wires M/M 200mm
- 1x Power supply 3-12V / 0,6A
- 1x Cable USB A - USB B - 1,8m

The Arduino Motor-Shield is stacked on top of the Arduino Uno R3 (Figure 5). The Arduino itself if connected with an external power supply to provied the motors with enough energy. For the PC-Arduino communication the Arduino is connected by using a USB A-B cable.

LEDs require an resistor to avoid damaging them. Basend on the source voltate of the Arduino (+5V) and the diodes forward voltage and current a matching resistor must be taken. In case of the white LEDs 150-220 Ohm would be sufficient.

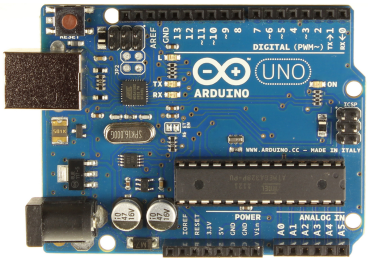

Figure 5: Arduino Uno R3

8 Lego construction

The converyor belt is build to fit the size of the card as seen in Figure 6. To provide a smooth movement of the belt serveral wheels are put in the frame.

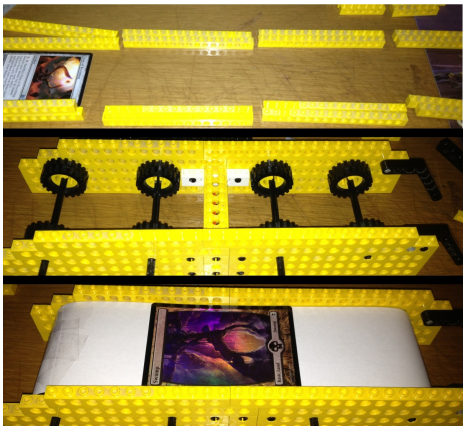

Figure 6: Converyor belt

The second part of the construction covers the box that will hold the cards. On the bottom of the box a hole is kept to attach the motor and a wheel. On the top a second wheel is attached to exert pressure from above (Figure 7).

Both parts are connected in a way to modify the angle of the throwout (Figure 8). This ensures the optimal angle according to the power of the motor. Special thanks for this idea goes to Christoph Petzoldt[22] who worked on a similar problem and provided me some hints.

[22] `http://standby-engineering.weebly.com`

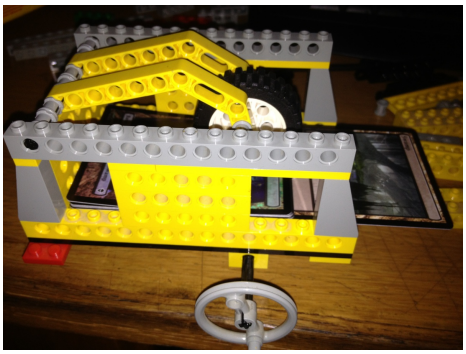

Figure 7: Cardstack

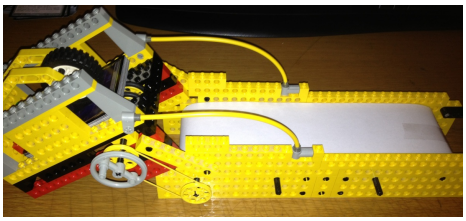

Figure 8: Converyor belt with cardstack

9 Conclusion

The final version and some videos can be seen on `http://www.spech.de/project/mtganalyser` and the authors website at `http://www.spech.de`.

The figures 9, 10 and 11 show the whole construction including the lego frame, the webcam and MTG Analyser.

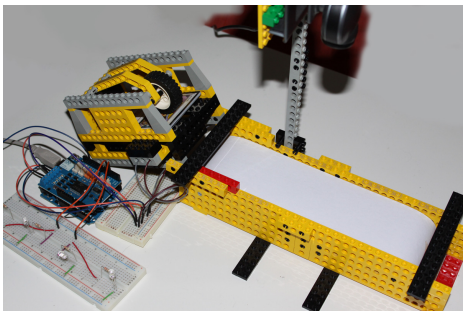

Figure 9: Final hardware with webcam

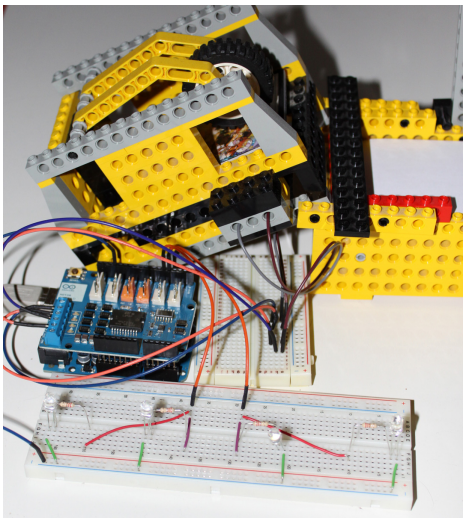

Figure 10: Arduino with LEDs and motors

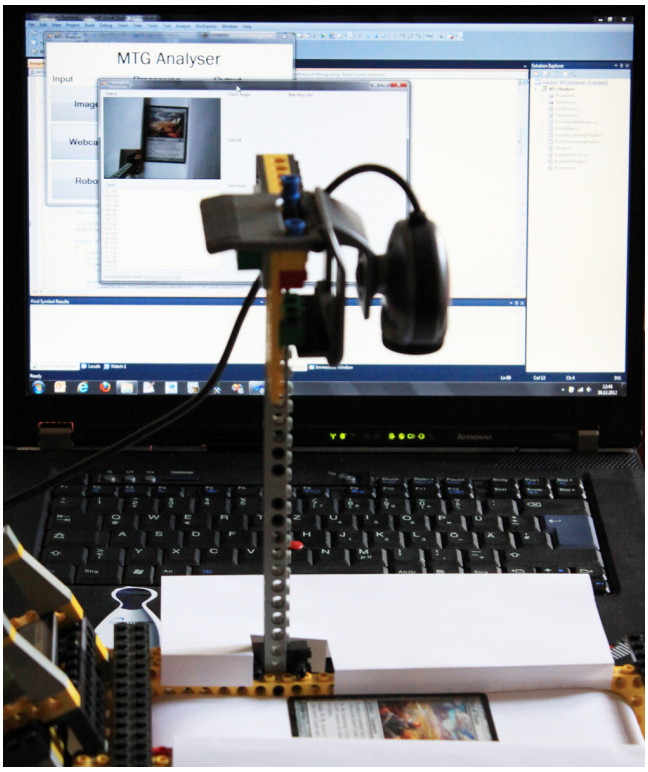

Figure 11: MTG Analyser