

# Early Computer Music Experiments in Australia and England

PAUL DOORNBUSCH

Australian College of the Arts, 55 Brady Street, South Melbourne, VIC 3205, Australia  
Email: pdoornbusch@collarts.edu.au

**This article documents the early experiments in both Australia and England to make a computer play music. The experiments in England with the Ferranti Mark 1 and the Pilot ACE (practically undocumented at the writing of this article) and those in Australia with CSIRAC (Council for Scientific and Industrial Research Automatic Computer) are the oldest known examples of using a computer to play music.**

**Significantly, they occurred some six years before the experiments at Bell Labs in the USA. Furthermore, the computers played music in real time. These developments were important, and despite not directly leading to later highly significant developments such as those at Bell Labs under the direction of Max Mathews, these forward-thinking developments in England and Australia show a history of computing machines being used musically since the earliest development of those machines.<sup>1</sup>**

## 1. INTRODUCTION

Music has always employed the latest technology to its ends. For example, the technology of drawing metal for strings made it possible to replace catgut for greater brilliance and volume, nineteenth-century precision manufacturing techniques of the day were used to develop instruments such as the pianoforte, and in the twentieth century there was the explosion of developments related to electrification. Music has always taken maximum advantage of available technology.

Today it may seem obvious to use computers in all aspects of music; from the creation of ideas and creating the music, to the performance, recording and reproduction process, computers are a ubiquitous and integral part of the practice. However, in the early days of computing, the machines were incredibly primitive by today's standards and turning their powers to music was a leap few would have made.

In the 1940s, the calculations required by the advancements in modern physics were unwieldy and tedious. To help overcome this, calculating machines

had been developed, such as the 'linear equations machine', the 'differential analyser' and the 'multi-register accounting machine' (Hemstead and Worthington 2005: 110; Hally 2006: 11–13). However, the calculating machines still required significant human intervention, so there was a desire to build an automatic calculator with some sort of system to store the data and the instructions for what to do with the data.

There were some major technological advances at the time that allowed the realisation of an automatic calculator with memory. One such advance was the use of thermionic valves (vacuum tubes), as switching devices. The other major advance was storage or memory systems such as mercury delay line 'memory' and the Williams Tube. Mercury delay line memory had been used in radar systems during the Second World War to process radar echoes. This memory system could be adapted for use in an automatic calculator, such as the Electronic Delay Storage Automatic Computer (EDSAC) or the CSIR Mark 1. Other early computers used different storage mechanisms, such as the Williams Tube, which used the phosphor of a cathode ray tube to store digital data.

The first *automatic* computers were developed to run calculations too cumbersome for humans with calculators to manage,<sup>2</sup> and also to organise and sort large volumes of data. With this fundamental function in mind, using a computer for music at all seems a rather odd idea. Indeed, the first book on computer music, *Experimental Music* by Hiller and Isaacson published in 1959, several years after the developments discussed in this article, says in chapter one:

Upon first hearing of the idea of computer music, a person might ask: 'Why program a digital computer to generate music?' The answer to this question is not simple, since such an undertaking immediately raises fundamental questions concerning the nature of musical communication

<sup>1</sup>The definition of a computer here means an all-electronic calculating machine with memory for the data and program; sometimes called a stored-program machine. This definition is expanded upon later, but the Manchester Small Scale Experimental Machine (Baby) from mid-1948 is generally considered to be the first such machine. There were several electronic and electromechanical programmable calculators in the 1940s (e.g. Colossus, the Zeus machines and the original ENIAC), which do not quite make the definition.

<sup>2</sup>The term 'automatic computer' may seem tautologous now, but in the late 1940s and early 1950s the term 'computer' referred to the *person* who operated a calculator. In that era, the computing department was a room full of people (often young ladies) running computations with calculators. When the first programmable electronic computers came along, they were called 'automatic' computers, to distinguish them from the person.

and its relation to formal musical structures. (Hiller and Isaacson 1959: 1)

Early computers produced more raw *sound* than anything else: they did not have the sort of display we are used to today, and they had electromechanical peripherals producing large amounts of sound in operation. So perhaps using a very early computer to play music was a more obvious and natural development because of the often-rhythmical sounds from the operation, and more easily imagined than using one to create an image.

The interface of early computers was nothing like today's computers. There was no alphanumeric keyboard for inputting information, and output was usually punched paper tape, transferred to another machine to print out text. There was little visual feedback beyond status lights, and sometimes a view into the memory on a cathode ray tube.

Several early computers included a loudspeaker used to communicate where the program was during execution. For example, a sound could be programmed to let the operator know the program had completed, or other sounds could be programmed to keep track of the execution of the program. Some of the computers that included a loudspeaker were the Ferranti Mark 1, Pegasus and Pilot ACE in England, UNIVAC-I and the TX-0 (at MIT) in the USA, CSIRAC<sup>3</sup> in Australia, the and others (Miyazaki 2012).

The audio interface of early computers was used as a signalling or diagnostic aid for users. A programmer would have the loudspeaker output a sound (called 'blurts' by CSIRAC programmers) indicating a program had completed or what stage had been completed. Early computers were not very reliable and sometimes produced erroneous results. Programmers would run a program multiple times for an important calculation.

The poor reliability of early computers meant that sometimes maintenance engineers and programmers could *hear* a problem before it was noticed by other means or erroneous output. One of the main engineers of the BINAC and UNIVAC-I computers, Louis Wilson, recalled that in 1949, to assess the state of the machine he would listen to interference created on a radio. He recalls:

When we were testing BINAC out, we were working two shifts and we worked all night long, and we had a radio going. After a while we noticed that you could recognize the pattern of what was happening at the moment by listening to the static on the radio. So I installed a detector in the console and an amplifier and a speaker so that you could deliberately listen to these things. (Miyazaki 2012)

<sup>3</sup>CSIRAC was originally known as the CSIR Mark 1 in Sydney, its name was changed to CSIRAC when in Melbourne in 1955 (Doornbusch 2004). The names are used interchangeably, but I have endeavoured to use CSIR Mark 1 for events from the Sydney time and CSIRAC for other times.

This is not a unique occurrence. Frank Cooper, an early computer engineer in England, recalls an event from the early 1950s when a commissioning engineer was sent to commission a new Ferranti Mercury computer:

He was in his hotel room waiting to go onsite to do some commissioning of a new Mercury, and he switched his bedroom radio on, and what should come on but an interview at the Bracknell weather centre: 'This super new computer [...] is now about to forecast the weather!' And lo and behold you heard some raucous semi-music coming out of this computer, which was supposed to be the computer forecasting the weather. However, Geoff, being very familiar with the machine, recognised this music as the overall test program, he also recognised that there was a fault in the multiplier of the computer. (Burton 1995)

Such diagnostic listening has been part of human behaviour for centuries. From the earliest medical practice to machine operators and technicians, people have used diagnostic listening to hear if there is an abnormality; a mechanic will *listen* to the machine in operation to determine what may be wrong. The practice of using audio output as a diagnostic and computer operation aid was not uncommon (Miyazaki 2012) as there were few other sensory interfaces, so it was perhaps more of a natural development that curious or creative programmers would try taking this further and make a computer play music.

## 2. EARLY DEVELOPMENTS IN THE USA

The documented history of computer music is dominated by the activity at Bell Telephone Laboratories in the USA. By the end of the 1950s, the USA was a country about 20 years out of the Great Depression and a decade after the Second World War, with a booming economy. This created an environment where speculative research could thrive. At the New Jersey Bell Telephone Laboratories in 1957, Max Mathews and John Pierce started experiments with making a computer synthesise waveforms. Utilising the first commercial digital-to-analogue converter (DAC), Mathews could synthesise any waveform (Roads 1996: 10); digital synthesis is theoretically unlimited and has unparalleled precision. However, the most powerful machines took many hours of processing to calculate a few seconds of sound waveform 'samples'. The output was digital magnetic tape, which could then be used with the DAC (Roads 2015: 93–4).

Mathews and Pierce applied the fundamentals of analogue synthesis to digital synthesis, which had applications in telephony. Pierce was a music enthusiast who realised that understanding how much *information* was in speech, and music, was useful for telephony research. Mathews, after going to a concert with Pierce, started writing the Music I program to perform music on the computer through digital waveform synthesis

(Park 2009). The first people to use the program were engineers. Mathews realised that composers would make full use of the extended abilities, so he solicited for composers to use the program. The composer James Tenney was employed to work in the Audio Research Department (Roads 1980). The requirements of composers spurred the development of more advanced MUSIC programs, culminating in MUSIC V, which is the historical precursor of the current computer music program Csound, and which was given away by Bell Telephone Laboratories to universities to advance the development of computer music.

While digital waveform synthesis and research into computer music began in the USA in the late 1950s, this was not the first time computers were used to play music. In Australia the CSIR Mark 1 was programmed to play music from about late 1950, and in England a Ferranti Mark 1 computer played music from about late 1951. Recently reported is activity from 1949 in the USA (Stuutz 2015) that may or may not have occurred. There were some similarities and differences in what happened in the various locations. However, none of the early groups had access to a DAC, and the computers would have barely been capable of waveform synthesis if a DAC was available. Also, the activity earlier than that in Bell Labs was not done with the intent of exploring what a computer might do for music, but rather as a programming challenge, entertainment, or a trick.

Betty Holberton is reported to have programmed the BINAC (Binary Automatic Computer) computer to play music in 1949 (Stuutz 2015), although this is doubtful. Another reported recollection of Holberton programming BINAC to play music in 1948 (Kleiman 1997) must be an error – BINAC ran its first test program in February 1949, but she mentions the delivery party may well be August 1949, which is what Holberton says in an oral history transcript (Tropp 1973: 169). Nancy Stern wrote the authoritative history of BINAC in several books and documents, she describes the Northrop Acceptance on 22 August 1949 in detail but does not mention it playing music (Stern 1979: 12). Given the detail and thoroughness in Stern's accounts, it would be an unusual oversight. Louis Wilson used radio interference as a diagnostic aid in 1949 when working on BINAC (Miyazaki 2012). An Associated Press article about BINAC on 22 August 1949 reports it calculating quickly (Altschull 1949: 7), however, they do not report BINAC playing music, which would be an obvious, spectacular event. Later in the report, Eckert says a *future* computer (UNIVAC) *may* play chess and write music as they can be described mathematically; he discussed the *future possibility* not what BINAC had done. Without evidence it is impossible to prove or disprove Holberton's claim. Possibly BINAC played music around 22 August 1949, but there are no other reports of this event and there were good opportunities for it to have been reported.

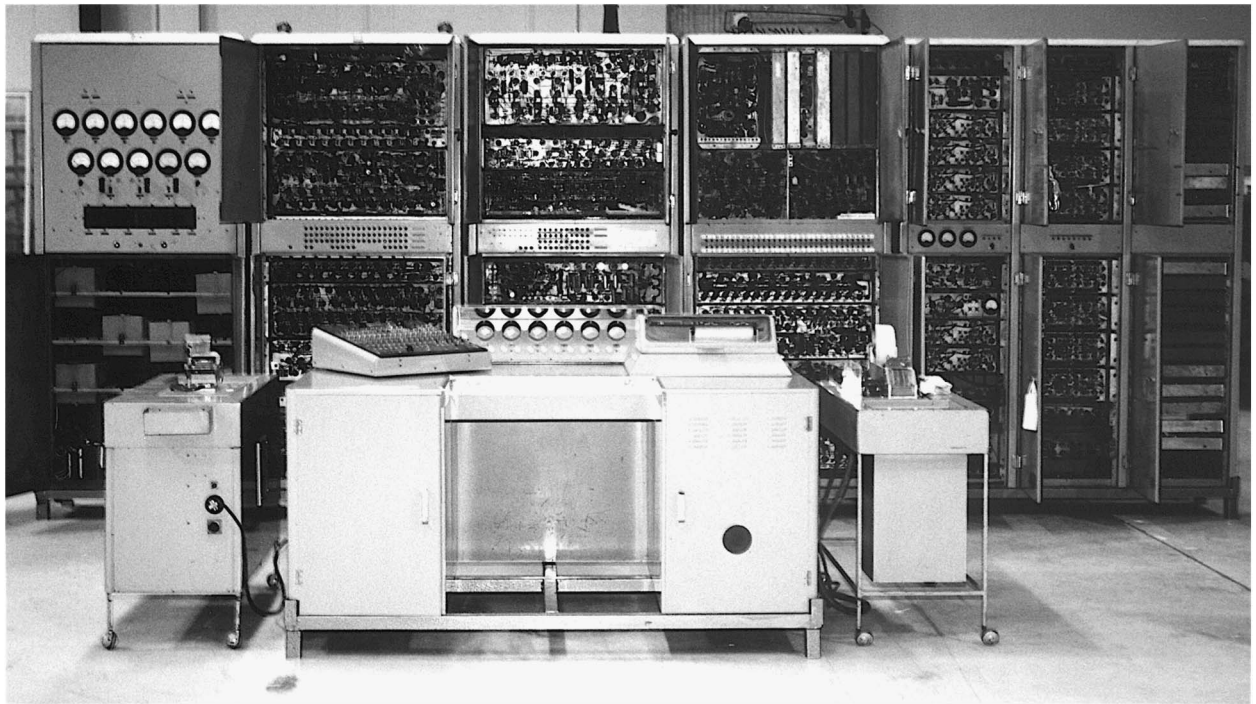
### 3. EARLY DEVELOPMENTS IN AUSTRALIA FROM 1950

In the late 1940s, Australia's Council for Scientific and Industrial Research (CSIR) embarked on a project to build a fully electronic digital computer. Trevor Pearcey, an English radio physicist, and Maston Beard, a researcher at the CSIR Radiophysics Laboratory in Sydney, designed the machine known then as the CSIR Mark 1 (later known as CSIRAC) and it ran its first program in November 1949. The CSIR Mark 1 was a serial computer designed as a proof-of-concept prototype to be commercialised as a more powerful machine. The first programmer was a mathematician called Geoff Hill, who had assisted with the logical design of the machine. Crucially, Hill came from a very musical family, and he had perfect pitch. He programmed the CSIR Mark 1 to play music from about late 1950.

As a *serial* computer, the CSIR Mk1 was thus very different from today's computers (Figure 1). A serial computer will transfer data from 'sources' to 'destinations'; for example, to and from memory, one bit at a time. Modern computers move data around in parallel chunks, typically 32 or 64 bits at a time. This difference in the architecture of the CSIR Mark 1 (and other similar machines) had significant consequences for programming the machine and it is especially problematic for applications with critical timing such as playing music in real time.

While developments in Australia occurred in isolation, the personnel involved were not cut-off from the rest of scientific advances. Pearcey had worked in England on advanced radar systems from 1940 to 1945, becoming well acquainted with the contemporary (electro) mechanical calculators. After the Second World War he travelled via North America to Australia. While in America he had contact with researchers, some of whom were using large university calculators in their work. He also had a chance meeting with Howard Aiken, who had completed a machine called the Automatic Sequence Controlled Calculator (ASCC, also known as the Harvard Mk1) for IBM at Harvard University. The ASCC was a very large complex calculator, using relays and partially controlled by a punched paper tape. In early 1947, Trevor Pearcey and Maston Beard officially commenced the 'Electronic Computer' project. Also in 1947 the University of Pennsylvania published the details of ENIAC.

The overriding considerations for the design of the CSIR Mark 1 were engineering and programming simplicity, as this was to be a prototype for a larger, more capable machine. The designers decided on a relatively small 20-bit word length, a serial architecture and a moderate instruction-clock frequency of 1000Hz. The first program, a simple multiplication of two numbers, was run late in 1949, probably in November but nobody recorded the exact date. Trevor



**Figure 1.** CSIRAC on display for its 50th birthday celebrations in Museum Victoria. The hooter loudspeaker can be seen in the right-hand door of the console. Image copyright the author.

Pearcey recalls, ‘We all shouted “Hooray!” and went back to work’ (Doornbusch 2005: 3).

### 3.1. The Hoot instruction and the loudspeaker

The CSIR Mark 1 had a loudspeaker attached to one of the serial ‘destinations’ and a ‘hoot’ instruction to alert the operator when the computer needed some sort of attention. By sending a string of pulses to the speaker, a sound would be made to signal an operator.

It is unclear exactly how Beard and Pearcey conceived of the loudspeaker. The original design documents do not show it, but at the end of 1948 Pearcey travelled to England and saw computers there, including the Manchester Mark I which had some sort of hooter, and EDSAC which had a bell when the STOP instruction was executed. Some first-generation computers had a non-programmable hooter; an instruction would trigger a fixed sound. Why Pearcey gave the CSIR Mark 1 a programmable hooter is a mystery and not documented anywhere (Doornbusch 2005: 19, 20). The Manchester Mark 1 did not initially have a programmable hooter: the records show the ‘hoot’ instruction first appeared in the evolving instruction set in February 1950 (NAHC 1951b; Lavington 2016). Turing’s programming manual for it mentions how a loop of instructions could be used to make a sound (Turing 1950: 24), however, there is no record of the Manchester Mark 1 being used to play any music, and Turing’s manual is from 1950.

It is probable that Pearcey recognised the need for a ‘hoot’ instruction, and implemented the simplest and most pragmatic solution of providing a programmable *destination* with a speaker attached. It is likely he saw the hooter on the Manchester Mark 1 (even if it was not programmable in 1948), and he simply copied the idea without much thought, as a simple, obvious, and direct solution. Pearcey planned for a programmable hooter on the CSIR Mark 1, without realising it offered the flexibility for a programmable frequency output, if one could negotiate the programming gymnastics to achieve it.

As a serial computer, the CSIR Mark 1’s speaker was built into the computer as a *destination* for data, effectively on a register of the machine and it received the raw pulse data off the ‘bus’. A single pulse would produce barely a click, for audible output multiple pulses would be needed. The notation used to send pulses to the speaker was the ‘P’ statement, and the programmers would notate this as ‘->P’. Several inline ‘P’ statements could be put in sequence or they could be organised as a short loop of instructions. The timing of the pulses (and of the loop) would have caused a change in the frequency of the sound from the hooter. Any programmer with an interest in sound or music would see the potential. However, there are two difficulties with this: the major problem to be overcome was the variable amount of time each memory access took of the mercury acoustic-delay-line memory. The other difficulty was the major clock frequency of CSIRAC, only 1000Hz, and this granularity

of instruction execution made generating pulses with exact timing very difficult. The sequence of instructions could, for example, look like this:

```

start of note
  set counter to x
  loop start, send pulse to speaker
    wait time t1
    send pulse to speaker
    wait time t1
    send pulse to speaker
    wait time t2
    send pulse to speaker
    wait time t3
    send pulse to speaker
  decrement counter
  if x equal to zero then exit, else
  return to loop
end of note

```

Achieving a steady stream of pulses to the speaker took considerable effort, a very intimate understanding of the machine's operation, and the arrangement of data and instructions in memory. Geoff Hill's musical knowledge was critical for him to program the CSIR Mark 1 to play music. Being a mathematician, he would know if he could send the hooter pulses with a regular period, it would make a steady pitch. Later, if the pulse-periods could be organised enough, it could play the notes of a diatonic scale.

Initially, playing music on the CSIR Mark 1 was a tremendous programming challenge. Naturally, Hill and the others programmed the CSIR Mark 1 to play popular melodies. They were engineers and scientists not specialist composers. The first song on the CSIR Mark 1 'Music Program' tape (a punched-paper tape), after a major scale, was Colonel Bogey, the popular wartime song. There are several references putting the first music on the CSIR Mark 1 as late 1950 or early 1951. Trevor Pearcey has been interviewed on video saying the music was a very early programming exercise in 1950 or 1951 (Doornbusch 2005: 24). Dick McGee, in an interview with the author, also remembered the music at the computing conference on 7–9 August 1951, but also in April 1951 when he started with the CSIRO (Doornbusch 2005: 25). There are other personal stories placing the CSIR Mark 1 music in very early 1951 or late 1950. This is strong evidence for putting the CSIR Mark 1 among the first computers to play music. Significantly, this happened six to seven years before the developments at Bell Labs, and it was real-time computer music unlike the work at Bell Labs. However, there were developments in England at about

the same time, and at least talk of something similar in the USA.

To the engineers, programming the CSIR Mark 1 to play music was a technical challenge in computation; the programmers were not investigating what this meant for creativity or the future of music. Sadly, it could have been otherwise. Percy Grainger was one of Australia's more adventurous composers with an interest in advanced music and he came very close to the CSIRAC when it was at the University of Melbourne in 1957. The staff of the computation laboratory would watch him walk past and say, 'There goes Percy Grainger!' (Doornbusch 2005: 82). Unfortunately they never thought to show him CSIRAC playing music and discuss the possibilities with him.<sup>4</sup> Grainger had developed Free Music Machines from the mid-1940s with rudimentary electronic synthesis techniques and had an acknowledged interest in music and the possibilities afforded by technology. The involvement of musicians (composers) with engineers is one of the crucial points of difference with the later developments in the USA and the great advances there.

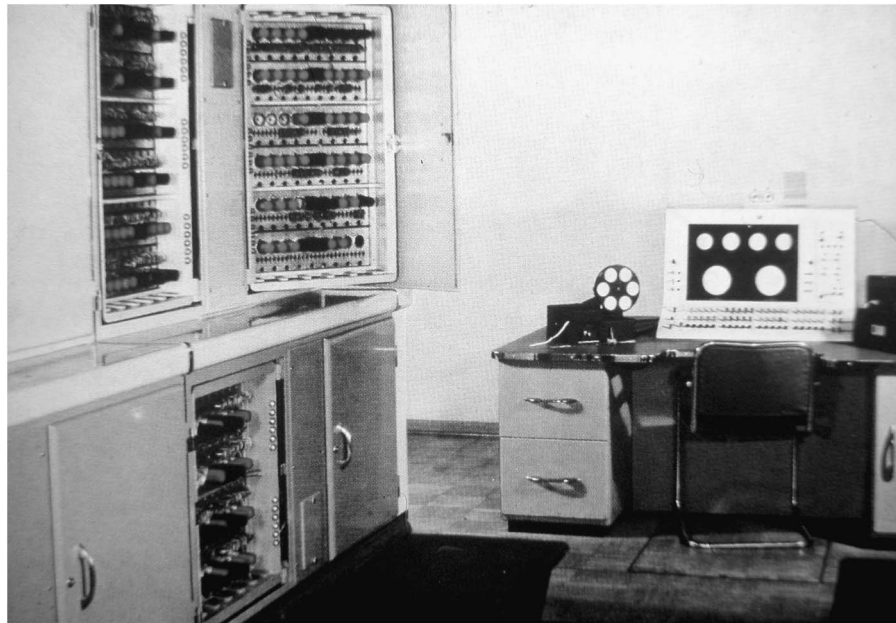
#### 4. EARLY DEVELOPMENTS IN ENGLAND FROM LATE 1951

Modern computing began in England in the mid-1940s, with the earliest machine recognisable as a computer by a modern definition<sup>5</sup> being the Small Scale Experimental Machine (SSEM) at the Victoria University of Manchester. The ideas spread quickly to other parts of the world where there was a need to perform large volumes of calculations or manage large amounts of data. Many early computers had similarities (such as the serial architecture, or mercury delay-line memory) dictated by the availability of technology to perform a particular task.

While the early computing developments in England are well documented, the music programmed on the early machines is practically unknown and undocumented. The Small Scale Experimental Machine (SSEM, operational mid-1948) in Victoria University of Manchester was further developed and improved to become the Manchester Mark 1 (operational mid-1949). Ferranti commercialised this design in 1951 as the Ferranti Mark 1 computer (Figure 2), and it was the world's first commercially available general-purpose computer (Allison 2003).

<sup>4</sup>Sound examples of the accurately reconstructed CSIRAC music (it was never recorded) are available online at [www.doornbusch.net/CSIRAC/index.html](http://www.doornbusch.net/CSIRAC/index.html).

<sup>5</sup>The history of computing shows a *continuum* of developments for several hundred years, and the distinction between a programmable calculator and a computer can be fine. However, from a modern perspective, a computer (as described by Turing in 1936 and von Neumann in 1945) is an all-electronic digital machine, with electronic memory containing data *and* instructions, and with a branching instruction.



**Figure 2.** Ferranti Mark 1 computer with some of the many cupboards open and console. The hooter loudspeaker was to the side of the console and cannot be seen. Image courtesy of the University of Manchester.

Christopher Strachey, research physicist at STC during the Second World War who's father worked with Alan Turing as a cryptographer, had an interest in computing. A schoolmaster after the Second World War, Strachey was given a copy of the 1950 programming manual for the Ferranti Mark 1 by his friend Turing (Burton 1995; Hally 2006: 100). In 1951 Strachey developed a program to write love letters by picking random words; Strachey's signature first appeared in the logbook on 25 September 1951 (NAHC 1951a; Lavington 2016). He also wrote a program to play draughts, completed in October 1951, and at the end of the program it played the national anthem, 'God Save the King'. Soon after, other programmers wrote programs to play music, the tunes played were 'Baa Baa Black Sheep', 'In the Mood', and so on. It was this program the BBC recorded and is recognised as the first recording of a computer playing music.

Turing's 1950 programming manual mentions the hoot instruction and the possibility of making a note 'rich in harmonics' with a pitch about middle C (Turing 1950: 24). However, Turing did not describe the process of making the Manchester Mark 1 or the Ferranti Mark 1 play music, he only described the 'hoot' instruction as a way to signal an operator. Strachey, took the next step to make the Ferranti Mark 1 play a steady note and melody, thus doing the same work Geoff Hill did in Australia to get the machine to play a recognisable melody.

In September 2016, researchers in New Zealand reported they had restored the recording made of Strachey's program playing music (France-Pressé 2016). The existence of the recording is not new

information; the BBC reported it in 2008 (Fildes 2008), it is also mentioned in 1995 (*Resurrection the Computer Conservation Society UK bulletin*, issue 12) and 2005 (Doornbusch 2005: 21). The recording was on a 78 rpm acetate disk (a vinyl 'master') and the report discusses the cleaning of this recording of the inevitable crackles and pops from the analogue recording format, to hear the original recording closer to how it would have sounded in late 1951. Such restoration is a standard software function now. There are many inaccuracies in a *Guardian* article: it claims the music was, 'Played on a gigantic contraption built by the British computer scientist Alan Turing.' However, it was played on a Ferranti Mark 1, as noted by Frank Cooper, who was there, and it was not created by Turing despite what the headline says (Burton 1995, 2016). The article also says Turing was a musical innovator and makes other similar claims, but there is no evidence for such claims and it is Christopher Strachey who took the steps to use the 'hoot' instruction to create music. To illustrate what Turing achieved, Turing's *entire* documentation of the hooter in the 1950 *Programming Handbook for Manchester Mark 1 Electronic Computer* (which also suited the Ferranti Mark 1) is reproduced here:

## 9.2 The hooter

When an instruction with function symbol /V is obeyed an impulse is applied to the diaphragm of a loudspeaker. By doing this repeatedly and rhythmically a steady note, rich in harmonics, can be produced. This is used to enable the operator to be called to attend to the machine in some way. The simplest case is where the whole of a job is

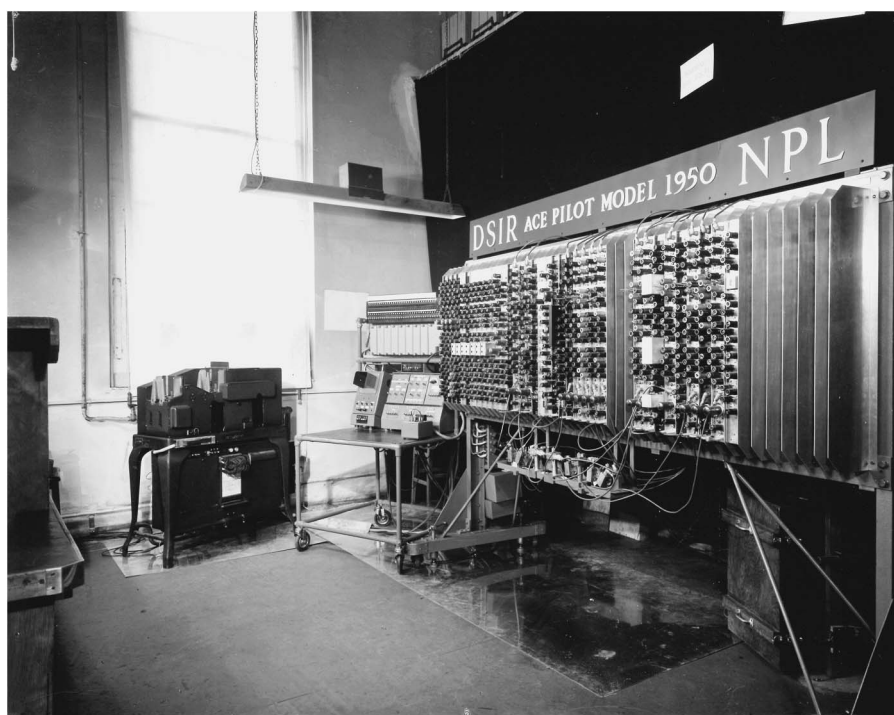


Figure 3. Pilot ACE computer. Image courtesy of the NPL.

completed and it is required to clear the electronic stores and start something different. All that is then required is to repeat a cycle of instructions including a hoot, e.g.

FS NS/V

CS FS/P

In this case every second instruction will put a pulse into the speaker. These pulses will occur at intervals of 8 beats i.e. 1.92 ms giving a frequency of 521 cycles (about middle C). Or one could use the loop of three instructions

O@ /V

G@ P@/V

M@ G@/P

which gives a slightly louder hoot a fifth lower in frequency. Single pulses applied to the loudspeaker are distinctly audible as something between a tap, a click, and a thump. This fact can be turned to good account. By putting hoot instructions into programmes at suitable points one is enabled to 'listen in' to the progress of the routine. Some indication of what is going on is given by the rhythm of the clicks that are heard. (Turing 1950)

This description could be said to *allude* to using the hooter to play at least musical notes, but only as signals about the machine's operation. As musical intervals are ratios, and Turing was a savant mathematician, it is also reasonable that Turing understood the timing of the instruction loop and expressed the ratios of the two frequencies as a musical interval in common usage.

The description avoids the complexities required to produce a scale (the finely tuned delays between the pulses), and Strachey made the leap to make the machine play a musical scale and produce recognisable melodies.

Listening to the music played by the Ferranti Mark 1 (Burton 1995), and reading Turing's manual, it would appear the sound production method was the same as on CSIRAC. The Ferranti Mark 1 used different memory technology to CSIRAC, the Williams Tube, which may have had improved access times, but this is currently unknown and worthy of further research. Williams Tube memory is random-access whereas delay line memory is sequential-access, and timing critical applications are inherently easier to program with random-access memory. The memory technology contributes greatly to the programming difficulty in achieving a steady and accurate pitch from the hooter, as does the main clock frequency of the machine.

A recording of the Ferranti Mark 1 computer playing music, along an interview in which Frank Cooper discusses the recording, may be found online.<sup>6</sup>

In another significant early development in England, unreported at least in the electronic and computer music communities, the National Physical Laboratory's (NPL) Pilot ACE computer (operational in mid-1950) played music autonomously from around 1952

<sup>6</sup>Recording of Ferranti Mark 1: [http://curation.cs.manchester.ac.uk/digital60/www.digital60.org/media/mark\\_one\\_digital\\_music/index.html](http://curation.cs.manchester.ac.uk/digital60/www.digital60.org/media/mark_one_digital_music/index.html). Interview with Cooper: [http://curation.cs.manchester.ac.uk/digital60/www.digital60.org/media/interview\\_frank\\_cooper/index-2.html](http://curation.cs.manchester.ac.uk/digital60/www.digital60.org/media/interview_frank_cooper/index-2.html) (accessed 17 January 2017).

(Figure 3). Designed by Alan Turing, the Pilot ACE was the prototype machine for the full ACE (Automatic Computing Engine), and it had a programmable hooter similar to the Ferranti Mark 1. Mercury acoustic delay line memory was used on the Pilot ACE, giving it variable instruction timing, similar to CSIRAC.

Most interesting is that the Pilot ACE had program routines that would allow it to create its own music. Davies reports:

The ACE Pilot Model and its successor, the ACE proper, were both capable of composing their own music and playing it on a little speaker built into the control desk. I say composing because no human had any intentional part in choosing the notes. The music was very interesting, though atonal, and began by playing rising arpeggios: these gradually became more complex and faster, like a developing fugue. They dissolved into coloured noise as the complexity went beyond human understanding. (Davies 1994: 19)

Diagnostic routines were built into the Pilot ACE, some sent pulses to the speaker. One diagnostic tool was a console meter triggered by data transfer instructions, used as a crude measure of program complexity and machine efficiency (fewer data transfers meant greater efficiency). However, the signal driving the meter had a higher frequency AC component obscuring the average reading, so Davies filtered this out to make the meter readings clearer, and out of curiosity sent it to the speaker. This was not initially intended to make music, but rather as a diagnostic aid, so programmers and operators could have an indication of efficiency. This was a kind of diagnostic listening as previously discussed, but it was also a kind of sonification of the machine's activity.

Another engineer, David Clayden added switches to aid in program testing and debugging, allowing the internal data or instructions to be changed. While testing this new facility the engineers accidentally discovered the ability to trick the machine into making music by overwriting memory. Davies recalls:

The program would be in a loop and therefore give a note from the speaker. By fixing the operation to add into a suitable bit in the instruction delay line, this counter would spill over into the timing fields and periodically enter a new kind of loop. Using the right switch settings made the loop size stay constant for a while (a note), then abruptly change in size.

Loops were always multiples of 32 microseconds long, so notes had frequencies which were submultiples of 31.25 KHz. The music was based on a very strange scale, which was nothing like equal tempered or harmonic, but was quite pleasant.

Initially the playing was rather slow, as though it was simply familiarising the listener with its unorthodox scale. Higher up the scale, when the notes are a bit high even for

young ears, the spilling over process gets more complex. It is next to impossible to work out what is going on in the delay line, with the instructions changing all the time. The next bars of music are a little faster and the same scale of notes appears, generally rising but now varied in sequence like an arpeggio. This repeats with ever faster and more complex patterns until it becomes too fast and complex to follow. Once you get used to the strange scale it really sounds musical and even intelligent.

We were amazed when we first heard it. To get a nice composition you had to choose the right setting on David's switches, and the settings for the interesting music were rather critical. Some switch settings gave falling arpeggios. (Davies 1994: 20)

This would appear to have been in the early 1950s, approximately 1952–53, as the Pilot ACE was shut down in 1955, and the full ACE machine was operational in 1957. Davies describes this event just as the computer was being readied to move to a new home as a service computer.

This seems to have been a happy accident, rather than an early planned exercise in computer-aided algorithmic composition. However, this seems to have happened before the computer-aided composition experiments by David Caplin and Dietrich Prinz in the mid-1950s (Ariza 2011), and quite some time before the highly documented and rigorous work of Hiller and Isaacson.

## 5. DISCUSSION

It seems like a small distinction, but these computers did not just play music on their own, the personnel who programmed them and used them *made* the computers play music. The programming required was extraordinary; it was convoluted and an order of magnitude more difficult than what was required for other computation tasks. Humans enjoy a challenge, and this was a programming challenge par excellence; to send regularly timed pulses to a speaker from machines with variable and unreliable timing for many things, including memory access. So it is extraordinary programmers were *bothered* (also with the Bell Labs developments, but at least there it may have had applications to their main business). The programmers in both Australia and England must have had a great interest in music (as they did in the USA), and also in the machines, to develop programs to play music. It was an extraordinary effort.

In Australia in the very early 1950s, the CSIR Mark 1 playing music was initially a programming challenge, it was also seen as a diversion and a *trick* of the computer (Doornbusch 2005: 25). By the late 1950s, the music played by CSIRAC was seen as an amusement and people were more impressed with its calculating abilities (Doornbusch 2005: 45). In a young

country focused on primary production, the music was never seen as something worthy of serious investigation. Also, when the CSIR Mark 1 was in Sydney (1949–55) it was part of the CSIR Radiophysics department, which was undergoing a realignment of proprieties. The CSIR Mark 1 was always intended to be a prototype machine to be commercialised outside the CSIR, thus concluding the project. The main personnel involved with the CSIR Mark 1 were trying to popularise it through the music to save the project. CSIR management of course resisted this, so the music played by the CSIR Mark 1 was never recorded or broadcast. The music developments with CSIRAC were not seen as leading to any productive activity, so they were not pursued any further beyond the novelty value and programming challenge of making a computer play music.

There is good evidence of the CSIR Mark 1 playing music either late in 1950 or very early 1951, and there is also good evidence that Christopher Strachey programmed the Ferranti Mark 1 to play music slightly later, probably mid-September 1951. Strachey's music output was recorded in November of 1951, and the CSIR Mark 1 played music publically in September 1951, but there was activity before these public events. The music played by the Ferranti Mark 1 in England was also seen as a *trick* (Burton 2016). The music was unrelated to the 'serious' business of performing calculations for advanced physics and engineering. Similarly, the engineers and management of the Pilot ACE did not see the musical output as worthy of further investigation, which is understandable in a country still recovering from the Second World War.

In light of the foregoing, it is unsurprising that the early computer music developments in Australia and England did not lead to any further investigation such as those from the early work at Bell Labs. The history of computer music celebrates the developments in the USA, mostly by Max Mathews, with justification. The early work in Australia and England is often overlooked, despite the fact that computer music activities there began some six years earlier than in the USA, and they played music in real time; something that did not happen in the USA until many years later.

There are several reasons for celebrating the USA developments over those in Australia and England. The main reason is that the work in the USA was designed from the outset to explore the possibilities of computer music and led to numerous future developments, while in Australia and England it was an amusement only. The early developments in Australia and England occurred before the availability of a DAC, so there was no flexibility in the sound production and thus less obvious potential. The only group that would have seen this as useful would have been advanced or experimental composers, but they were not involved and unaware of the potential.

Digital waveform synthesis with a DAC allows any waveform to be synthesised with excellent control and precision, with obvious potential for sound synthesis regardless of how costly it was of computer resources and time. In America, Max Mathews realised engineers were not excellent musicians and, crucially, he advertised for composers to become involved with the project. In an interview in 1980, he recalls:

When we first made these music programs the original users were not composers, they were the psychologist Guttman and John Pierce and myself, who are fundamentally scientists. We wanted to have musicians try the system to see if they could learn the language and express themselves with it. So we looked for adventurous musicians and composers who were willing to experiment. The first one was David Lewin, who was at Harvard at the time. We corresponded and he did a composition mostly by mail, which was a brave thing to do. Then John Pierce met Jim Tenney at the University of Illinois, where Tenney was studying with Hiller. Pierce was very much impressed with Tenney's music and his interest in computers. He invited Tenney to take a temporary job at the Laboratories to try out the music programs. Tenney took the job and developed some timbres of his own and also some pieces. (Roads 1980)

Clearly, Bell Telephone Laboratories (at least Pierce's group) was the sort of company to see potential and value in such a speculative venture. To meet composers' needs, Mathews developed software with additional flexibility and functionality. However, this did not happen in England and Australia, where the developments occurred too early to use DACs and such speculative research was unsupported. Bell Labs started the first ever project with the *intent* of exploring the possibilities of computer music and a composer was employed to work with the researchers. These combined factors had the effect of keeping the developments in Australia and England unknown for decades; they did not lead to further developments and in modern terms the results were very unsophisticated. The early work of Christopher Strachey in England is *still* not well researched or documented, and is an area for further investigation.

There are other developments in electronic music from Australia and England that are also largely ignored, such as the film sound experiments of the early 1930s by Jack Ellitt and the electronic music machines and pieces by Percy Grainger. Ellitt and Grainger chose to follow their experimentalist dreams by emigrating (to Europe and America respectively) where they could be with more like-minded individuals and engage in their practice. Grainger at least made regular trips 'home' and attempted to engage with his original culture. Perhaps these people and their work are less known because they did not inspire further developments, or because they occurred in a cultural context where such radical ideas could not take hold.

## 6. CONCLUSION

The history of electronic and computer music is made of many small and often uncoordinated steps; the developments discussed above are some of those steps. While these developments did not lead to the enormous advancements that happened at Bell Labs under the direction of Max Mathews, they are nonetheless noteworthy. The developments in England and Australia show that, at the very least, people seem to have a natural predisposition to turn *anything* new to the production of music. When discussing the CSIR Mark 1 developments from 1950 with Max Mathews in 2000, Mathews wrote:

I found it very interesting and the implications about human beings most encouraging to me, the most impressive thing was the shortness of the time between the development of a computer powerful enough to make any kind of musical sound and the actual production of music with the computer. I think this shows how fundamental and important music is to human beings. (Mathews 2000)

When we first chatted about the CSIRAC music, Mathews instantly asked about the memory architecture of CSIRAC, and when I told him, he groaned and said how ‘hard’ it would be to get a steady note from CSIRAC; he *immediately* understood the implications and the difficulty of the task.

While in Australia and England the first steps were taken to make a computer play music, and also to create music independently, there was not the technology for arbitrary waveform synthesis (via a DAC) or the cultural environment to see the potential benefits of this. Australia in the early 1950s was focused on primary industry, and technology was viewed as either supporting that end or it was unnecessary. Similarly, England was emerging from the Second World War at the time and focused on rebuilding a shattered country. Thus, CSIRAC or the Ferranti Mark 1 or the Pilot ACE playing music was a fun ‘trick’ and a useful programming challenge or tool, but it was not seen as impressive as the raw calculation ability of the machines and it could not be seen as a *useful* undertaking for societies either building a new country or rebuilding an old one.

## ACKNOWLEDGEMENTS

I am indebted to several people who have helped to ensure the accuracy of this article: Chris Burton and Simon Lavington of the UK Computer Conservation Society, also Cort Lippe, Curtis Roads and Peter Thorne. I am very grateful for their generous assistance, which has helped to improve this article and ensure its accuracy.

## REFERENCES

- Allison, J. 2003. Introduction to the Mark 1. <http://curation.cs.manchester.ac.uk/computer50/www.computer50.org/Mark1/Mark1intro.html> (accessed 29 August 2016).
- Altschull, H. 1949. New Electronic ‘Brain’ Plays Chess, Even Writes Music. *Pittsburgh Post-Gazette*, 22 August. [www.newspapers.com/newspage/87777548/](http://www.newspapers.com/newspage/87777548/) (accessed 12 November 2016).
- Ariza, C. 2011. Two Pioneering Projects from the Early History of Computer-Aided Algorithmic Composition. *Computer Music Journal* 35(3): 40–56.
- Burton, C. 1995. Interview with Frank Cooper. [http://curation.cs.manchester.ac.uk/digital60/www.digital60.org/media/interview\\_frank\\_cooper/interview\\_frank\\_cooper.mp3](http://curation.cs.manchester.ac.uk/digital60/www.digital60.org/media/interview_frank_cooper/interview_frank_cooper.mp3) (accessed 9 June 2016).
- Burton, C. 2016. Personal email communication with author.
- Davies, D. 1994. Very Early Computer Music. *Resurrection, the Bulletin of the Computer Conservation Society* 10. [www.computerconservationsociety.org/resurrection/pdfs/res10.pdf](http://www.computerconservationsociety.org/resurrection/pdfs/res10.pdf) (accessed 13 September 2016).
- Doornbusch, P. 2004. Computer Sound Synthesis in 1951: The Music of CSIRAC. *Computer Music Journal* 28(1): 10–25.
- Doornbusch, P. 2005. *The Music of CSIRAC: Australia’s First Computer Music*. Melbourne: Common Ground.
- Fildes, J. 2008. ‘Oldest’ Computer Music Unveiled. London, BBC. <http://news.bbc.co.uk/2/hi/technology/7458479.stm> (accessed 30 September 2016).
- France-Presse, A. 2016. First Recording of Computer-generated Music – Created by Alan Turing – Restored. *The Guardian*. [www.theguardian.com/science/2016/sep/26/first-recording-computer-generated-music-created-alan-turing-restored-enigma-code](http://www.theguardian.com/science/2016/sep/26/first-recording-computer-generated-music-created-alan-turing-restored-enigma-code) (accessed 26 September 2016).
- Hally, M. 2006. *Electronic Brains: Stories from the Dawn of the Computer Age*. London: Granta Books.
- Hempstead, C. A. and Worthington, W. E. (eds.) 2005. *Encyclopedia of 20th-Century Technology. Volume 1*. New York: Routledge.
- Hiller, L. A. and Isaacson, L. M. 1959. *Experimental Music*. New York: McGraw-Hill.
- Kleiman, K. 1997. Singing BINAC 1948. CYHIST Community Memory: Discussion List on the History of Cyberspace. <https://groups.yahoo.com/neo/groups/cyhist/conversations/messages/1271> (accessed 7 November 2016).
- Lavington, S. 2016. Personal email communication with author.
- Mathews, M. 2000. Personal email communication with author.
- Miyazaki, S. 2012. Algorithmic Listening 1949–1962. Auditory Practices of Early Mainframe Computing. AISB/IACAP World Congress 2012, Birmingham, UK, 2–6 July 2012 Symposium on the History and Philosophy of Programming.
- NAHC 1951a. Mark I Log Books, 1951–58, 6 vols. UK National Archive of the History of Computing, Manchester. Catalogue number NAHC/MUC/2/C6.
- NAHC 1951b. Miscellaneous Reports 1951. UK National Archive of the History of Computing, Manchester. Catalogue number NAHC/MUC/2/C9. A single typed sheet dated 28 February 1950.
- Park, T. H. 2009. An Interview with Max Mathews. *Computer Music Journal* 33(3): 9–22.

- Roads, C. 1996. *The Computer Music Tutorial*. Cambridge, MA: MIT Press.
- Roads, C. 1980. Interview with Max Mathews. *Computer Music Journal* 4(4): 15–22.
- Roads, C. 2015. *Composing Electronic Music: A New Aesthetic*. Oxford: Oxford University Press.
- Stern, N. 1979. The BINAC: A Case Study in the History of Technology. *Annals of the History of Computing* 1(1): 9–20.
- Stuutz, U. 2015. Computer Music in 1949? Ip's Ancient Wonderworld blog. <http://irrlichtproject.blogspot.com.au/2015/11/computer-music-in-1949.html> (accessed 7 November 2016).
- Tropp, H. S. 1973. Computer Oral History Collection, 1969–1973, 1977. Smithsonian Museum. [http://amhistory.si.edu/archives/AC0196\\_bart730427.pdf](http://amhistory.si.edu/archives/AC0196_bart730427.pdf) (accessed 7 November 2016).
- Turing, A. 1950. *Programming Handbook for Manchester Mark 1 Electronic Computer*. Alan Turing Archive. [www.alanturing.net/turing\\_archive/archive/m/m01/M01-030.html](http://www.alanturing.net/turing_archive/archive/m/m01/M01-030.html) (accessed 19 August 2016).